

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky  
a komunikačních technologií

BAKALÁŘSKÁ PRÁCE

Brno, 2017

Šimon Vicen



# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

## FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

## ÚSTAV TELEKOMUNIKACÍ

DEPARTMENT OF TELECOMMUNICATIONS

## WEBOVÝ PORTÁL PRO PODPORU VÝUKY

WEB PORTAL FOR SUPPORT OF EDUCATION

### BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

### AUTOR PRÁCE

AUTHOR

Šimon Vicen

### VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. Jiří Schimmel, Ph.D.

BRNO 2017

# Bakalářská práce

bakalářský studijní obor **Teleinformatika**  
Ústav telekomunikací

**Student:** Šimon Vicen

**ID:** 177293

**Ročník:** 3

**Akademický rok:** 2016/17

**NÁZEV TÉMATU:**

## Webový portál pro podporu výuky

### POKYNY PRO VYPRACOVÁNÍ:

Pomocí redakčního systému WordPress realizujte webový portál pro podporu výuky, který bude mít sekci veřejnou a neveřejnou, dostupnou pouze studentům vybraných studijních oborů a zaměstnanců VUT po přihlášení VUT Loginem. Administrace portálu bude od tohoto přihlášení zcela oddělena. Portál bude podporovat fotogalerie, odběr novinek, diskusní fórum, kalendář akcí a databázi studijních materiálů. V tomto portálu realizujte také vazbu na úložiště souborů, a to včetně možnosti nahrání a stažení celých složek souborů - projektů, doplněných o informace o autorovi, názvu, aplikaci, pro niž je projekt určen, datu vytvoření a komentář, které zadá uživatel ručně při nahrávání souborů na úložiště nebo formou textového souboru, který bude obsažen v nahrávaném adresáři.

### DOPORUČENÁ LITERATURA:

- [1] McNULTY, S., WordPress - efektivní publikování na webu. Zoner Press, 2009. 256 s. ISBN 978-80-7413--42-7
- [2] KOSEK, J., PHP - tvorba interaktivních Internetových aplikací. Praha: Grada, 1998. 492 s. ISBN 80-7169-3-3-1

**Termín zadání:** 1.2.2017

**Termín odevzdání:** 8.6.2017

**Vedoucí práce:** doc. Ing. Jiří Schimmel, Ph.D.

**Konzultant:**

**doc. Ing. Jiří Mišurec, CSc.**  
*předseda oborové rady*

### UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

## **ABSTRAKT**

Táto práca sa venuje problematike tvorby webovej stránky na podkladoch redakčného systému wordpress s prihlasovaním SSO cez prihlasovanie na stránkach VUT. Práca v postupných krokoch popisuje atribúty SSO a spôsoby, akými môžeme tento cieľ dosiahnuť a zrealizovať vo webových službách. Ďalej sa práca venuje funkciám štandardu federácie eduid.cz a technológiám, ktoré jej slúžia. V praktickej časti sa práca venuje návrhu stránky, webovým aplikáciám, uložiškom prístupným pre užívateľov a funkčnému prihlasovaniu cez stránky VUT s evidovaním prihlásených užívateľov.

## **KĽÚČOVÉ SLOVÁ**

Metadáta, MySQL, SAML, Server, SSO, PHP, Wordpress, Webový server

## **ABSTRACT**

The thesis is focused on creation of web page based on wordpress development system in work with single sign-on login system via VUT web site in Brno. Thesis stepwise explains SSO attributes and ways how we can achieve this goal to make it work within given web services. Moreover, the thesis also explains functions of eduid.cz federation and technologies that it works with. Practical part is dealing with design of web, web applications, front-end cloud storage accessed by users and page working with login system via VUT sites and registering its users.

## **KEYWORDS**

Metadata, MySQL, SAML, Server, SSO, PHP, Wordpress, Web server

VICEN, Šimon. *Webový portál pro podporu výuky*. Brno, 2017, 55 s. Bakalárska práca. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedúci práce: Ing. Jiří Schimmel, Ph.D.

## VYHLÁSENIE

Vyhlasujem, že som svoju bakalársku prácu na tému „Webový portál pro podporu výuky“ vypracoval samostatne pod vedením vedúceho bakalárskej práce, využitím odbornej literatúry a ďalších informačných zdrojov, ktoré sú všetky citované v práci a uvedené v zozname literatúry na konci práce.

Ako autor uvedenej bakalárskej práce ďalej vyhlasujem, že v súvislosti s vytvorením tejto bakalárskej práce som neporušil autorské práva tretích osôb, najmä som nezasiahol nedovoleným spôsobom do cudzích autorských práv osobnostných a/alebo majetkových a som si plne vedomý následkov porušenia ustanovenia § 11 a nasledujúcich autorského zákona Českej republiky č. 121/2000 Sb., o práve autorskom, o právach súvisiacich s právom autorským a o zmene niektorých zákonov (autorský zákon), v znení neskorších predpisov, vrátane možných trestnoprávných dôsledkov vyplývajúcich z ustanovenia časti druhej, hlavy VI. diel 4 Trestného zákoníka Českej republiky č. 40/2009 Sb.

Brno .....

.....

podpis autora

## POĎAKOVANIE

Rád by som poďakoval vedúcemu bakalárskej práce p. Ing. Jiřímu Schimmelovi, Ph.D. za odborné vedenie, konzultácie, trpezlivosť a podnetné návrhy k práci.

Brno .....

.....

podpis autora

# OBSAH

<b>Úvod</b>	<b>10</b>
<b>1 Platforma webového portálu</b>	<b>12</b>
1.1 Wordpress . . . . .	12
1.1.1 wordpress.com . . . . .	12
1.1.2 wordpress.org . . . . .	13
<b>2 Webové služby a aplikácie</b>	<b>15</b>
2.1 HTML . . . . .	15
2.2 PHP . . . . .	16
2.2.1 PHP kód a jazyk HTML . . . . .	17
2.3 Apache server . . . . .	18
2.4 MySQL . . . . .	19
2.4.1 MySQL a PHP . . . . .	19
2.4.2 SMTP . . . . .	20
2.4.3 OAuth 2.0 . . . . .	20
<b>3 Jednotný prístup do webových aplikácií a služieb</b>	<b>22</b>
3.1 Známe štandardy na poskytovanie SSO . . . . .	22
3.1.1 Enterprise Single Sign-On . . . . .	22
3.1.2 Webové Single Sign-On . . . . .	23
3.1.3 Kerberos . . . . .	23
3.1.4 Federace Identity eduid.cz . . . . .	24
3.1.5 OpenID . . . . .	25
3.1.6 SAML . . . . .	25
<b>4 Výsledky študentskej práce</b>	<b>28</b>
4.1 Inštalácia wordpressu.org . . . . .	29
4.2 SAML a návrh realizácie prihlasovania . . . . .	31
4.2.1 Self-signed certificate . . . . .	31
4.2.2 Výmena metadát . . . . .	31
4.2.3 Nastavenie atribútov . . . . .	33
4.3 Návrh podporovaných aplikácií . . . . .	34
4.3.1 Fórum . . . . .	34
4.3.2 Odber noviniek . . . . .	34
4.3.3 Úložisko . . . . .	35
4.4 Programovanie modulov a aplikácií . . . . .	37
4.4.1 Function.php . . . . .	37

4.4.2	Wp-config.php . . . . .	37
4.4.3	Style.css . . . . .	37
4.4.4	Unzip.php . . . . .	37
4.4.5	Class.fileup.php . . . . .	37
<b>5</b>	<b>Záver</b>	<b>39</b>
	<b>Literatúra</b>	<b>41</b>
	<b>Zoznam symbolov, veličín a skratiek</b>	<b>43</b>
	<b>Zoznam príloh</b>	<b>44</b>
<b>A</b>	<b>Zdrojový kód aplikácie</b>	<b>45</b>
A.1	Wordpress súbory . . . . .	45
A.1.1	function.php . . . . .	45
A.1.2	wp-config.php . . . . .	45
A.1.3	style.css . . . . .	46
A.2	Subory webový aplikácií . . . . .	48
A.2.1	unzip.php . . . . .	48
A.2.2	class.fileup.php . . . . .	49
<b>B</b>	<b>Prístup na webovú stránku</b>	<b>54</b>
B.1	Povolenia na prihlásenie: . . . . .	54
B.2	Ako sa prihlásiť: . . . . .	54
B.3	Administrátorský prístup: . . . . .	54
B.4	Dočasne vytvorená emailová schránka . . . . .	54
<b>C</b>	<b>Obsah přiloženého CD</b>	<b>55</b>



## ZOZNAM OBRÁZKOV

1.1	Prostredie systému wordpressu.com . . . . .	13
1.2	Prostredie systému wordpressu.org . . . . .	14
2.1	HTML požiadavka na server (inšpirácia zo stránok jakpsatweb.cz) . .	18
2.2	Vytvorenie MySQL databázy . . . . .	19
2.3	Požiadavka PHP na server (inšpirácia zo stránok jakpsatweb.cz) . . .	20
3.1	VUT SAML Self-Signing Certificate . . . . .	26
4.1	Prístup k linuxovému serveru . . . . .	28
4.2	Prístup k wordpressu cez FTP . . . . .	29
4.3	Vytváranie databázy . . . . .	30
4.4	Návrh webového úložiska . . . . .	36
4.5	VUT stránka . . . . .	38

# ZOZNAM VÝPISOV

2.1	Ukážka PHP kódu . . . . .	17
4.1	Ukážka metadát SP . . . . .	32
4.2	Ukážka metadát IDP . . . . .	33
A.1	Function.php . . . . .	45
A.2	wp-config.php . . . . .	45
A.3	style.css . . . . .	46
A.4	Unzip.php . . . . .	48
A.5	class.fileup.php . . . . .	49

# ÚVOD

V tejto práci sa budem podrobnejšie zaoberať vytvorením webu prostredníctvom redakčného systému WordPress v kombinácii s programovacím jazykom PHP a implementáciou prihlasovania do systému na základe autentifikácie vybranej skupiny študentov a zamestnancov pomocou prihlásenia cez VUT login. V práci budem rozoberať výhody a nevýhody systému jednotného prihlásenia, rovnako ako jeho význam pre bezpečnosť a užívateľský komfort v prostredí internetu.

Internet ako celosvetový verejne dostupný systém vzájomne prepojených počítačových sietí prenášajúcich dáta zamieňaním paketových dát pomocou štandardizovaných protokolov, v prvom rade IP a mnohých ďalších, zaradzujeme do kategórie médií, bez ktorých si každodenný život nedokážeme predstaviť.

Rýchly pokrok internetu nezadržateľne rastie, no s ním vznikajú rafinovanejšie škodlivé softvéry, ktoré prelomia slabšie zabezpečenú sieť do pár minút. Preto sa bezpečnosť pomaly ale isto dostáva v posledných rokoch na prvé miesto. S narastajúcim množstvom škodlivého softvéru narastajú aj požiadavky v rámci bezpečnosti, napr. na minimálnu dĺžku, štruktúru, roznorodosť hesiel do rôznych aplikácií (veľké, malé písmená, čísla, špeciálne znaky atď.). Práve táto zložitosť je často kontra-produktívna. Preto vznikajú technológie, ktoré by ponúkli vyšší komfort, odľahčili používateľa a zároveň by zachovali stupeň bezpečnosti. Práve tieto riešenia ponúka technológia jednotného prihlasovania SSO.

V súčasnej dobe stále viac a viac vidíme jej využívanie, práve v dobe, kedy je potrebné využívať množstvo hesiel k najrôznejším systémom a aplikáciám. Jedná sa o technológiu sprostredkujúcu prístup k viacerým nezávislým systémom pomocou jedného prihlasovacieho údaju pod správou jednej z organizácií. V tejto práci podrobnejšie rozoberám technológie webových aplikácií, návrh a ich spracovanie, podporu webových služieb pre bezpečnú komunikáciu s e-mailovým serverom, a funkciou jednotného prihlasovania externých aplikácií na platforme eduid.cz na VUT v Brne.

Internet využívame na sprostredkovanie obsahu v podobe množstva dát, informácií s inými užívateľmi, či už formou blogu alebo webovej stránky. V dnešnej dobe existuje množstvo spôsobov, ako v relatívne krátkom čase vytvoriť a prispôbiť webovú stránku podľa vlastných potrieb a predstáv. Je na uvážení každého, ktorú z možností uprednostní.

Samotná webová stránka môže byť tvorená tromi spôsobmi:

1. redakčný systém,
2. pomocou wysiwyg editoru,
3. vytvorenie vlastnej stránky.

Bližšie sa vo svojej práci budem taktiež zaoberať tvorbou webu, možnosťami využitia moderných technológií a dostupných programovacích jazykov v prostredí redakčného systému WordPress.

# 1 PLATFORMA WEBOVÉHO PORTÁLU

## 1.1 Wordpress

Wordpress je open-source redakčný a publikačný systém CMS pre tvorbu blogov a webových stránok. Je vyvíjaný pod licenciou GNU GPL, čo zaručuje, že WordPress je dostupný bezplatne a je možné modifikovať kód podľa vlastných potrieb za predpokladu, že užívateľ poskytuje kód voľne ostatným členom komunity s rovnakou licenciou. Vďaka svojej širokej užívateľskej a vývojárskej komunite a rozsiahlej technickej podpore sa radí k najpoužívanejším OpenSource systémom, kde zaraďujeme taktiež aj webový software ako Joomla či Drupal. [1]

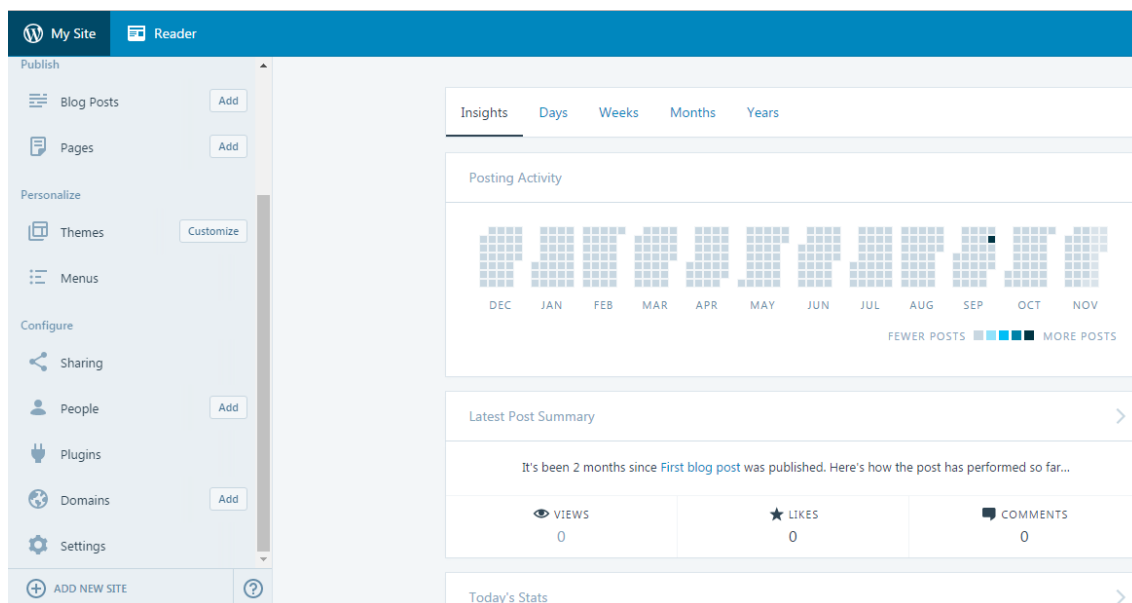
Myšlienka vzniku WordPressu prišla v roku 2003 ako snaha o vytvorenie dobre štruktúrovaného osobného publikačného nástroja, postaveného na jazyku PHP a MySQL. Jadro softvéru je vyvíjané stovkami dobrovoľníkov z komunity. WordPress ponúka množstvo modulov a pluginov, ktoré sú dostupné zdarma alebo za minimálny poplatok. WordPress je k dispozícii v dvoch verziách, záleží od užívateľa na čo stránku potrebuje a čo by stránka mala spĺňať. [1] [2]

### 1.1.1 wordpress.com

Server [www.wordpress.com](http://www.wordpress.com) vykonáva značnú časť používateľskej práce zaňho, keďže nie je potrebné hľadať vhodného poskytovateľa webhostingu, taktiež vykonáva prácu pri aktualizovaní softvéru, inštalácii jednotlivých komponentov, ako aj zabezpečenie servera, a teda užívateľ sa môže zamerať na blogovanie, resp. na textovú časť svojej stránky.

Ako vidíme na obr. 1.1, táto služba za to, že poskytuje služby, požaduje, aby súčasťou domény v URL odkaze bolo [wordpress.com](http://wordpress.com), čo nie je vhodné a lákave pre profesionálne tvorby webu (napr. pri tvorbe webu firiem).

Do úvahy treba taktiež zobrať aj vyššie náklady za využívanie pokročilejších funkcií a v neposlednom rade [wordpress.com](http://wordpress.com) neposkytuje prístup ku kódu blogu, čo znamená, že je obmedzená modifikácia jednotlivých modulov. Takže, keď to zhrnieme, môžeme konštatovať, že [wordpress.com](http://wordpress.com) sa skôr odporúča pri nenáročnom zdieľaní informácií na webe, ako je napr. cestovateľský blog. [3]



Obr. 1.1: Prostredie systému wordpressu.com

## 1.1.2 wordpress.org

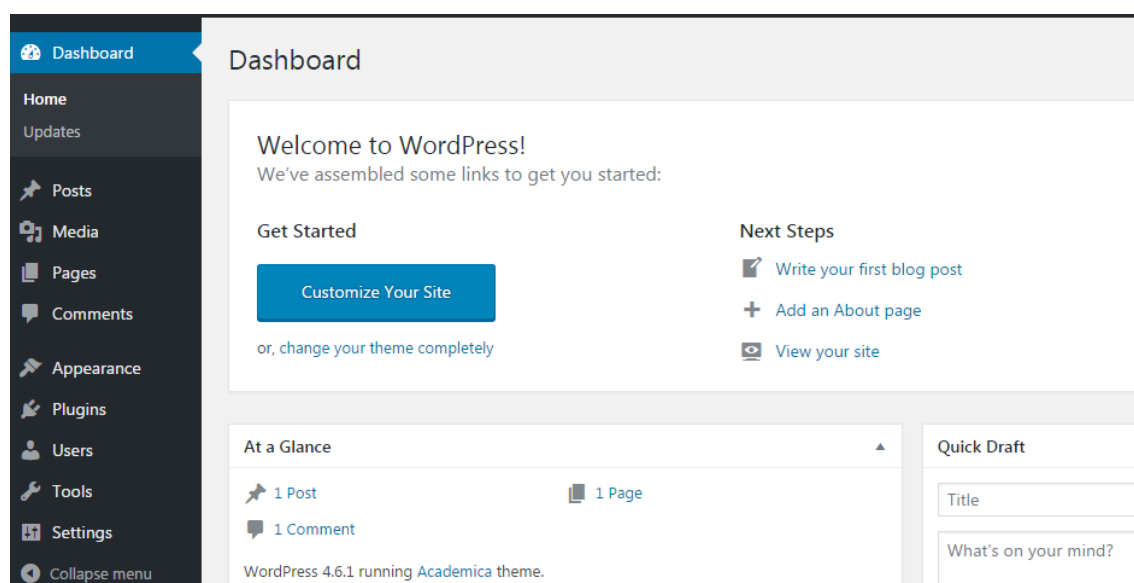
V prípade serveru [www.wordpress.org](http://www.wordpress.org) hovoríme o niečo komplexnejšej tvorbe webu. Proces spočíva v stiahnutí inštalačných súborov a následnom nainštalovaní WordPressu na vhodný, či už platený alebo neplatený webhosting.

Jedná sa o voľbu, kde má užívateľ väčšiu kontrolu nad webovou stránkou, má možnosť prispôbovať, upravovať kód, či kombinovať s inými programovacími jazykmi. V opačnom prípade, ako môžeme vidieť na obr. 1.2, treba počítať s tým, že užívateľ musí celú správu webu prebrať na vlastnú zodpovednosť. [3]

Ak si zvolíme platformu [wordpress.com](http://wordpress.com) je potrebné :

- udržiavať zálohu webu,
- korigovať web pri zvýšenej prevádzkovej náročnosti,
- v prípade komplikácií sám nájsť riešenie daného problému.

Pre spracovanie svojej práce som využíval práve túto formu redakčno-publikačného systému.



Obr. 1.2: Prostředí systému wordpressu.org

## 2 WEBOVÉ SLUŽBY A APLIKÁCIE

### 2.1 HTML

Pri zakladaní služby WWW postačovali pre fungovanie tri základné technológie:

1. **štrukturovaný jazyk HTML** - dodnes je neodeliteľnou súčasťou webu,
2. **HTTP** - veľmi dôležitý protokol, vďaka ktorému je zaistený prenos HTML-stránok zo vzdialeného serveru do prehliadača užívateľa,
3. **URL** - každý z objektov umiestnený na webe má svoju jedinečnú URL adresu, ktorá slúži na odkazovanie na daný objekt.

V súčasnej dobe už je spojenie týchto troch technológií je značne nepostačujúce, keďže maximálne, čo môže ponúknuť, je séria navzájom prepojených elektronických dokumentov. Z tohto pohľadu webové aplikácie prešli viacerými inováciami.

Prvá z podstatnejších inovácií viedla k vytvoreniu „generátora stránok“, čo znamená, že stránky sa automaticky menili pri zmene obsahu. HTML stránka je v princípe dokument uložený na disku servera. Inovácia spočívala v odkazovaní URL adresy na spustiteľnú aplikáciu, teda program, ktorý pohotovo vygeneruje tento zobrazovaný dokument. Vytvorením rozhrania CGI, ktoré definuje spôsob spustenia programu a predávania dát medzi serverom a programom, stránka mohla obsahovať aktuálne informácie bez väčšieho zásahu správcu.

Medzi ďalšie inovácie patrila úprava CGI skriptu, definovanie formulára či vytvorenie vyhľadávania kľúčových slov. CGI skript bol pôvodne napísaný v jazyku C a C++. Na úpravu boli používané ďalšie z radu jazykov, ako napríklad Pearl, neskôr použitie Javascriptu, ktorý sa zapisoval priamo do HTML. Najčastejšie išlo o využitie pre prácu s formulármi na strane servera, inak nazývaný aj SSJS. O niečo neskôr prišiel systém ASP od firmy Microsoft, ktorý na obsluhu využíval programovacie jazyky VBScript a JScript. Ako najvhodnejší sa ukázal práve PHP. Princíp použitia je obdobný ako pri SSJS a ASP, no je šírený „freeware“, čiže bezplatne. V nasledujúcich podkapitolách by som sa chcel venovať trojici programom nazývaných „Triáda“, ktoré sú dôležité pri tvorbe dynamických stránok.[4]



## 2.2 PHP

Väčšinová tvorba dnešných aplikácií sa sústreďuje na web. Oproti klasickým desktopovým aplikáciám majú niekoľko výhod:

- finančne výhodnejšie,
- značne uľahčená správa aplikácie (pri aktualizáciách na webe novú verziu stačí nahrať a okamžite ju môžu využívať všetci, ktorí o ňu majú záujem),
- je zaručená vyššia bezpečnosť (keďže informácie sú uložené na serveri a ak server neobsahuje bezpečnostné chyby, neoprávnení užívatelia sa k nim nedostanú),
- v bezproblémovej kompatibilite (keďže prístup je riešený cez webový prehliadač a nie operačný systém). [5]

PHP je jednou z alternatív pri tvorbe týchto webových aplikácií. Je to voľne šíriteľný programovací jazyk, s oficiálnou stránkou na adrese <https://secure.php.net/>, kde okrem samotných PHP inštalačných súborov vieme nájsť množstvo informácií vzťahujúcich sa k PHP, podporu pri komplikáciách, ktoré vzniknú užívateľom pri práci v PHP prostredí, či pravidelné softvérové aktualizácie. Jedná sa o programovací jazyk, ktorý pracuje na strane serveru.

Pôvodný význam zkratky PHP bol Personal Home Page Tools, ktorý bol neskôr skrátený na Personal Home Page (PHP). Vznikol v roku 1996, kedy si Rasmus Lerdorf vytvoril v programovacom jazyku Pearl jednoduchý systém pre evidovanie prístupu k vlastným stránkam. Od tej doby prešlo PHP veľkými zmenami. Boli pridávané rozšírenia, doplnenia o mnohé dokumentácie, možnosti prepojenia s databázovým systémom SQL či vytváranie formulárov a mnohé ďalšie.

Veľkú rolu pri preferovaní jazyka PHP hrá jeho nezávislosť na platforme, či už Unix alebo Windows, keďže nie je viazaný žiadnym konkrétnym serverom. Najlepšie sa však ukázala kompatibilita so serverom Apache, ktorý bol použitý aj na linuxovom serveri, pri tvorbe mojej práce. [4] [6]

## Výpis 2.1: Ukážka PHP kódu

```
<?php
require( dirname(__FILE__) . '/wp-load.php' );

if ( force_ssl_admin() && ! is_ssl() ) {
    if ( 0 === strpos($_SERVER['REQUEST_URI'], 'http') ) {
        wp_redirect( set_url_scheme( $_SERVER['REQUEST_URI'],
            'https' ) );
        exit();
    } else {
        wp_redirect('https://' . $_SERVER['HTTP_HOST']
            . $_SERVER['REQUEST_URI'] );
        exit();
    }
}

function login_header( $title = 'Log In',
    $message = '', $wp_error = '' ) {
    global $error, $interim_login, $action;

    add_action( 'login_head', 'wp_no_robots' );
    add_action( 'login_head', 'wp_login_viewport_meta' );
?>
```

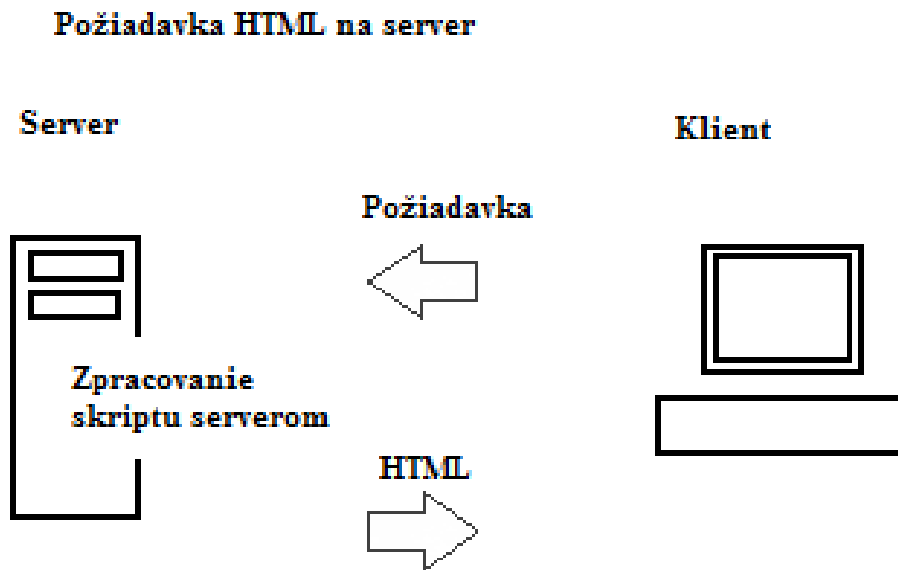
### 2.2.1 PHP kód a jazyk HTML

S rozšírením internetu prišla obrovská vlna webových statických stránok. Nejednalo sa o nič zložité, jednoduchý textový súbor, dokument s nadpisom, odstavcami a obrázkami určený len na čítanie. Pri návšteve jednoduchej HTML stránky prehliadač pošle požiadavku na server, server prijme požiadavku, spracuje ju, vyhodnotí o ktorý súbor sa jedná a zašle ho bez úprav naspäť, (znázornené na obr.2.1).

Pri návšteve PHP stránky sa odohráva jeden krok naviac. Po požiadavke od prehliadača server nájde potrebný súbor, no predtým, ako ho zašle naspäť, server spracuje zdrojové PHP súbory, na základe čoho sa vytvorí výstupné HTML, ktoré následne prehliadač zobrazí. Na serveri beží CGI skript, čo je vlastne program, ktorý dokáže vygenerovať užívateľovi do stránky, čo potrebuje. Stránka na serveri teda neleží staticky, ale je tvorená dynamicky, tj. prispôsobuje sa tomu, kto ju využíva. (znázornené na obr.2.3)

Všetky potrebné dáta su uložené v databáze a bežiaca webová aplikácia má svoje vlastné rozhranie, cez ktoré je možné pridávať nové objekty, prípadne ich upravovať. Ak by dáta neboli uložené v databáze, ale jednalo by sa o klasickú HTML stránku, bolo by nutné spravovať veľké množstvo stránok. Taktiež by nebolo možné vytvárať ďalšie skripty, rýchle vyhľadavanie dát, diskusné fóra, či iné užitočné moduly.

Ako príklad môže slúžiť zobrazenie aktuálneho času na webovej stránke. V PHP skripte vieme nastaviť čítanie kódu, ktoré ukazuje napr. na aktuálny serverový čas a dátum. PHP príkazy sú vložené do HTML kódu ako vsuvka, konkrétne do tagov „<?“ PHP a „?>“, ako môžeme vidieť vo výpise 2.1. Súbory s týmito vstupmi sú pomenované príponou .php. V prípade, že je požiadavka na vygenerovanie PHP skriptu, server prehľadáva celý súbor a obsah programovo vyhodnocuje, takže klientovi následne odošle už čisté HTML. [7]



Obr. 2.1: HTML požiadavka na server (inšpirácia zo stránok jakpsatweb.cz)

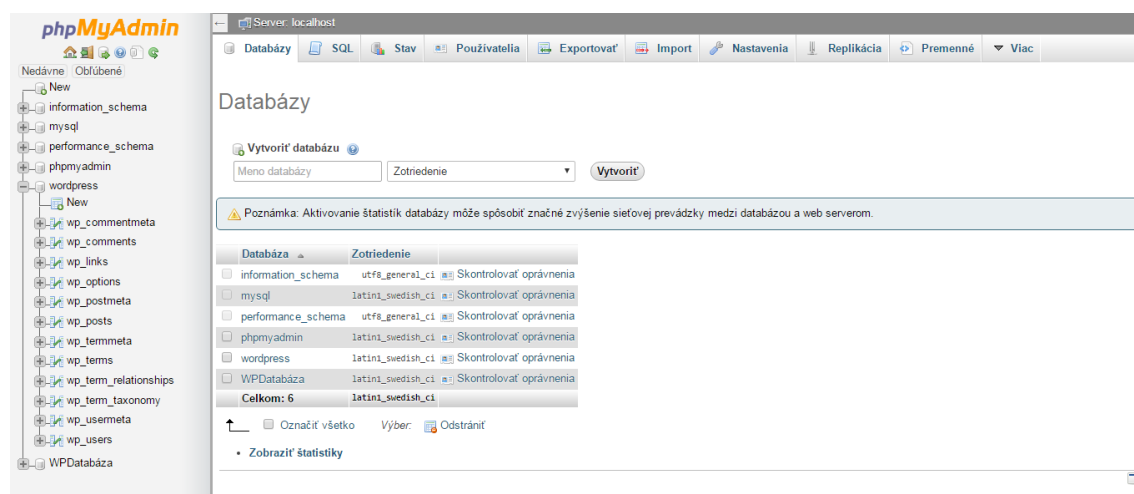
## 2.3 Apache server

Apache server je softwarový webový server s licenciou Open source, teda je dostupný zdarma. Je kompatibilný s platformou operačných systémov Linux či Windows. Využíva sa na väčšine domácich serverov, kde rolu hrá aj jeho veľmi rýchla a jednoduchá inštalácia.

Tak, ako Apache server podporuje množstvo programovacích jazykov či databáz, s ktorými môže užívateľ pracovať, podporuje taktiež rôzne formy autentizácie pre bezpečné fungovanie servera (napr. vykonávanie kompresí stránok alebo prepisovanie adries, čo je dôležité pri optimalizácii pre vyhľadávače) a vie vytvárať logy návštevnosti, kde sa ale jedná už o pokročilejšie formy webovej analytiky. [8]

## 2.4 MySQL

MySQL možno definovať ako Open-Source relačný databázový systém typu DBMS, ktorý vychádza zo základov programovacieho jazyka SQL. Jedná sa o jeden z najpoužívateľnejších databázových systémov, vďaka svojej licencií a rýchlemu prístupu k dátam. Má určité obmedzenia napr. oproti databáze Oracle, čo jej ale práve umožňuje fungovať rýchlejšie. MySQL je vo veľkej miere využívaná na ukladanie obrázkov, textu či iného obsahu, ku ktorým možno pristupovať, ďalej s nimi pracovať a modifikovať. [9]

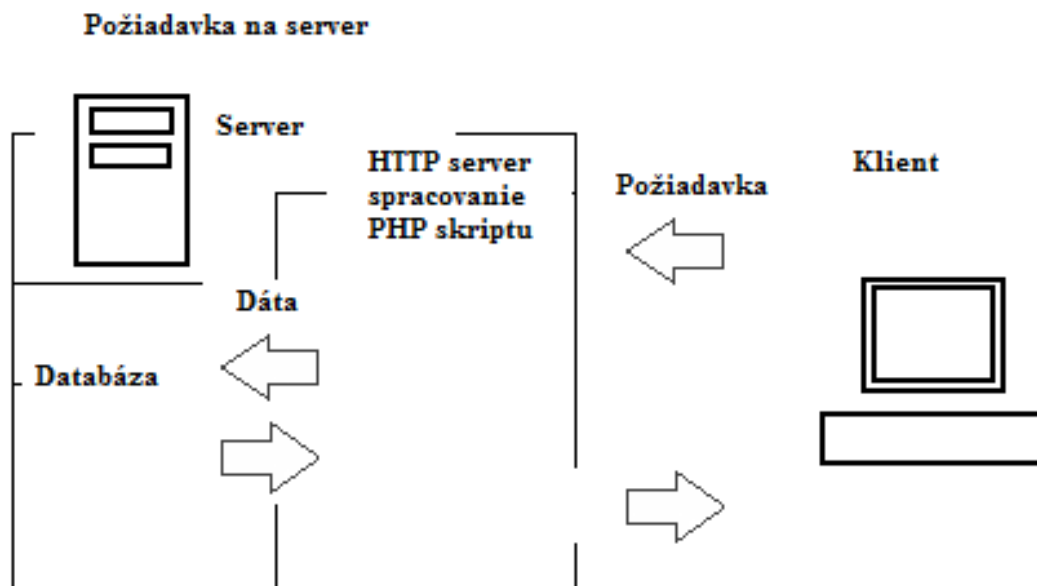


Obr. 2.2: Vytvorenie MySQL databázy

### 2.4.1 MySQL a PHP

Najčastejšie sa MySQL využíva v spojitosti s programovacím jazykom PHP, ktorý slúži na jednoduchý prístup k dátam.

Pre jednoduchú správu MySQL databáz sa využíva v súčasnosti široko využívaný nástroj nazývaný PhpMyAdmin. Je to pokročilý Open-Source program napísaný v jazyku PHP, ktorý slúži ako nástroj pre kompletnú správu MySQL systému cez webové rozhranie. Umožňuje používateľovi zálohovať, vytvárať tabuľky, vkladať, editovať či mazať záznamy v tabuľkách, rovnako ako vytvárať databázy. [10]



Obr. 2.3: Požiadavka PHP na server (inšpirácia zo stránok jakpsatweb.cz)

## 2.4.2 SMTP

SMTP je protokol, ktorý umožňuje jednoduchý prenos e-mailových správ medzi jednotlivými koncovými zariadeniami. Zaručuje úspešné doručenie pošty pomocou priameho spojenia medzi odosielateľom a adresátom. Posielaná správa je finálne doručovaná do poštovej schránky, do ktorej môže užívateľ pristupovať vďaka jednému z internetových protokolov POP či IMAP. Tie prijímajú elektronickú poštu zo vzdialeného servera prostredníctvom TCP/IP spojenia.

SMTP servery aj klienti smerujú e-maily na základe záznamu MX v DNS. Po komunikácii s DNS serverom sa preloží názov domény na IP adresu a správa je predaná e-mail serveru. SMTP server počúva na porte 25 protokolu TCP. Na port 25 taktiež smeruje všetky svoje odpovede, ktoré obdrží pri komunikácii s klientom. Pri využívaní a posielaní e-mailových správ je preto dôležité špecifikovať e-mailový server, ktorý sa postará o doručovací proces. Vykoná transfer pošty od klienta k inému koncovému zariadeniu. [11]

## 2.4.3 OAuth 2.0

OAuth definujeme ako Autorizačný Framework umožňujúci aplikáciám tretích strán získať a obmedziť prístup k HTTP službám. Je štandardom oproti iným autorizačným mechanizmom pri odstraňovaní problémov s overovaním totožnosti. OAuth 2.0 je možné použiť na zabezpečenie komunikácie medzi klientom a serverom. Pri klasickom klient-server overovaní totožnosti klient pošle požiadavku na server so svojimi

bezpečnostnými údajmi. Ak sa jedná o klientskú aplikáciu tretej strany, tá si uchováva uložené bezpečnostné údaje a pri každej ďalšej požiadavke na server ich zasiela, zvyčajne v textovej forme.

OAuth 2.0 funguje ako autorizačná vrstva, ktorá oddeľuje klienta od zabezpečeného zdroja na severi. Mechanizmus overenia totožnosti funguje tak, že prihlasovacie údaje, ktoré sú použité na prístup k zabezpečenému zdroju, sú iné, ako bezpečnostné údaje klienta. Klient dostane prístupový tzv. „access token“ po preukázaní svojej identity, ktorý ďalej používa na komunikáciu so zabezpečeným zdrojom. Prístupový token má obmedzenú platnosť a obsahuje informácie napríklad o rozsahu bezpečnostného oprávnenia. Komunikácia medzi klientom a autorizačným serverom môže prebiehať tiež pomocou digitálne podpísaných správ.

Ak to zhrieme, OAuth rieši autorizáciu – teda to, či má niekto právo pristupovať k určitým údajom. Pre riešenie autentifikácie (tj. identifikácie toho, či užívateľ je ten, za koho sa považuje) slúži rozšírenie OpenID Connect, čo je samostatná časť špecifikácie, ktorú daný používateľ môže a nemusí použiť. [12]

## 3 JEDNOTNÝ PRÍSTUP DO WEBOVÝCH APLIKÁCIÍ A SLUŽIEB

Princíp technológie SSO spočíva v tom, že je postačujúce, aby si užívateľ namiesto veľkého množstva prihlasovacích atribútov pamätal len jeden údaj. Všetko ostatné má na starosti systém SSO. Môžeme povedať, že SSO sa stalo bránou do systémov násobných, spojených, tak ako úplne nezávislých.

Dôvodom vzniku a existencie SSO je jednoduchý. Je to viditeľné na postupnom náraste požiadaviek v rámci bezpečnosti, na minimálnu dĺžku, štruktúru hesiel do rôznych aplikácií (veľké, malé písmená, špeciálne znaky atď. ). Práve táto zložitosť je často kontraproduktívna. SSO rieši túto problematiku uskutočnením automatického prihlásenia a užívateľ sa pohybuje v systémoch akoby pracoval v prostredí jednej aplikácie. Dochádza k zvýšeniu produktivity a komfortu, zníženiu času stráveného pri riešení problémov a vyhľadávani technickej podpory.

Ako príklad slúži platforma Windows, kde po prihlásení sa do počítača dochádza k overeniu identity voči zložke Active Directory <sup>1</sup>. Keď je následne potrebná autentizácia do iného systému využívajúceho účty na rovnakej doméne alebo je zaistené mapovanie účtov na doméne, využijú sa dáta, ktoré už sú uložené v počítači. [13]

### 3.1 Známe štandardy na poskytovanie SSO

- Enterprise Single Sign-On
- Webové Single Sign-On
- Kerberos
- Federace Identity
- OpenID
- SAML

#### 3.1.1 Enterprise Single Sign-On

Systémy, ktoré fungujú na báze jednotného prihlásenia typu Enterprise Single Sign-On, zabezpečujú prihlásenie v systéme a po prvom prihlásení sa zachytávajú výzvy sekundárnych aplikácií k zadaniu overovacích informácií a automaticky dopĺňujú požadované ID a heslo. [14]

---

<sup>1</sup>adresová služba implementovaná do systémov Windows na zaistovanie autentizácie a autorizácie užívateľov

### 3.1.2 Webové Single Sign-On

Webové SSO alebo Web access management sprostredkuje užívateľom koncových zariadení prístup k aplikáciám a ich zdrojom pomocou internetového prehliadača. Overovanie užívateľa prebieha pomocou archivácie identifikačných údajov v protokolových dátach, tzv. cookies, na internetovom proxy serveri, respektíve na internetovom serveri cieľovej adresy. Táto procedúra urýchľuje a zjednodušuje prístup používateľa k webovej adrese alebo zdroji objektu dosiahnuteľnému pomocou webového prehliadača. [15]

### 3.1.3 Kerberos

Kerberos definujeme ako sieťový autentizačný protokol, ktorý pracuje na základe tzv. ticketov<sup>2</sup> a využíva kryptografiu pre bezpečné overovanie tak ako klienta, tak server cez nezabezpečenú sieť, aby umožnil komunikujúcim stranám preukázať svoju identitu.

Na začiatku zadáme svoje prihlasovacie údaje a o všetko ostatné sa postará SSO. Varianta Kerberos je používaná ako autentizačná metóda pre Windows 2000, Windows XP a Windows Server 2003. [16]

#### Vlastnosti Kerberosu:

- užívateľské heslo ani jeho hašovacia funkcia sa nikdy neprenáša sieťou, iba jeho „tickety“, čo je jeho výhodou,
- je využívaná symetrická kryptografia (napr. DES, RC4, AES),
- štruktúra je modifikovateľná a rozširiteľná - je možné využiť ďalšie bezpečnostné metódy (šifry, hashe),
- je podporovaná technológia tzv. vzájomnej autentizácie, tak ako klient u služby, ale i služba klientovi,
- je chránená proti odpočúvaniu tretou stranou, „eavesdropping“,
- umožňuje užívateľom pristupovať k službám bez zadania hesla, Single Sign-On,
- zjednodušená administrácia, tj. všetky účty sú spravované centrálné,
- užívateľ nemá prístup priamo k zdroju, namiesto neho to vykonáva systém.

---

<sup>2</sup>klientský software prezentujúci prístup



### 3.1.4 Federace Identity eduid.cz

Systém EDUID je združenie organizácií sprostredkujúcich služby využívané na bezpečný prenos informácií, dát či poskytovania služieb a aplikácií v prostredí jednej siete. Cieľom platformy EDUID je na vzájomnej dohode, v rámci rešpektovania ochrany osobných údajov, poskytovať služby (napr. konektivitu, výpočetný výkon) či identity jednotlivých členov, resp. organizácií zapojených do projektu a splňajúcich podmienky registrácie. O správu celej federácie sa stará organizácia CESNET.

Aby bolo možné narábať s dátami v rámci EDUID prostredia a prihlásenie bolo do systému EDUID a aby prihlásenie do systému bolo úspešné, je potrebné požiadať operátora o zaradenie metadát do centrálnej databázy. Následne je potrebné zariadiť realizáciu potrebných komponentov, inštaláciu a nastavenie využívaného softvéru, na ktorom systém EDUID funguje.

Federácia Identity používa štandardy protokolov (SAML a WebSecurity) pre vyžiadanie povolenia a overenia identity užívateľov medzi aplikáciami a systémami. Taktiež umožňuje organizáciám poskytovať SSO medzi rôznymi sieťami.

Operátor federácie, v prípade federácie eduid.cz sa jedná o Cesnet, sa zároveň stará o beh federácie ako celku. Uskutočňuje registráciu členov, poskytuje podporu a rieši bezpečnostné incidenty. Organizácie sa stávajú členmi federácie ak úspešne prešli procesom registrácie. Následne môžu členovia organizácie poskytovať dáta o svojich používateľoch alebo poskytovať sieťové služby a aplikácie. [17]

Podľa toho delíme komponenty federácie na:

1. **poskytovateľov identít** (IDP) – napojené k používateľským databázam, vykonávajú autentizáciu a poskytujú informácie o používateľoch,
2. **poskytovateľov služieb** (SP) – poskytujú online služby a zdroje používateľom federácie.

#### Výhody pre používateľov poskytovateľov identít:

- svojím jediným používateľským menom a heslom spravovaným domovskou inštitúciou používateľa získajú prístup k radu sieťových služieb,
- správcovia aplikácií neudržia autentizačné dáta ani autentizáciu nevykonávajú,
- autentizácia používateľa prebieha vždy v kontexte domovskej organizácie, citlivé autentizačné údaje neopúšťajú domovskú sieť,
- federačná infraštruktúra poskytuje jednoduchý, štandardný a bezpečný spôsob výmeny informácií o užívateľoch.

### 3.1.5 OpenID

OpenID je ďalší z otvorených štandardov popisujúci spôsob, akým možno overovať prístup užívateľov. Na strane poskytovateľa služby nie je nutný systém na autentizáciu používateľov. OpenID má tvar unikátneho URL, ku ktorému sú priradené identifikátory a heslá. V rozhraní OpenID je implementovaná služba sprostredkujúca presmerovanie autentizácie identity správcovi daného OpenID účtu. Správca následne schváli alebo zamietne danú žiadosť.

OpenID pokrýva celú škálu oblastí autentizácie, od obecného mechanizmu, až po špeciálne prípady použitia portálmi ako napríklad : AOL, BBC, Google, IBM, MySpace, Orange, PayPal, VeriSign, LiveJournal, Yandex, Ustream či Yahoo!. V Českej republike poznáme zástupcov portálu Seznam.cz, tak ako veľmi významná implementácia OpenID pod pseudonymom mojeID, pod združením CZ.NIC. [18]

### 3.1.6 SAML

SAML je štandard dátového formátu založený na báze XML. Je nápomocný pri vymieňaní autentifikačných a autorizačných dát medzi účastníkmi, respektíve medzi poskytovateľom identít a poskytovateľom služieb.

V roku 2001 sa skupina OASIS rozhodla vytvoriť XML framework pre bezpečnú výmenu informácií o používateľoch v daných organizáciách a služba SAML sa tak poslednou aktualizáciou štandardizovala na verziu 2.0. Vývoj softvéru prebiehal z už existujúcich štandardov:

- Security Services Markup Language (S2ML) od Netegrity,
- AuthXML od Securant,
- XML Trust Assertion Service Specification (X-TASS) od VeriSign,
- Information Technology Markup Language (ITML) od Jamcracker.

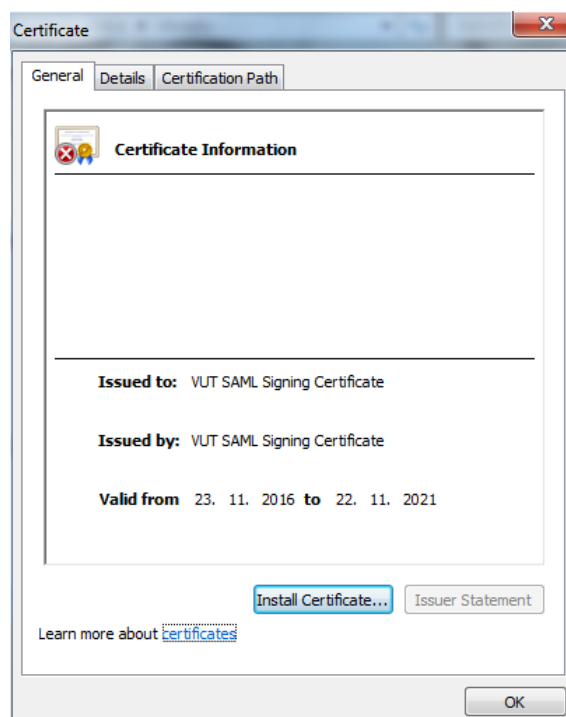
Jednou najdôležitejších úloh, okrem funkcie jednotného prihlasovania na webe, spojených s prijímaním, vysielaním a zdieľaním bezpečnostných informácií je poskytovanie XML formátov pre zabezpečované dáta používateľov a formátov na žiadanie a prenos informácií.

Medzi ďalšie neoddeliteľné súčasti SAML funkcií možno zadať určovanie ako tieto správy majú pracovať s protokolmi ako napr. SOAP. SAML poskytuje taktiež definovanie výmeny správ pre bežné použitie ako web SSO, podporu množstva mechanizmov na zabezpečenie súkromia či určenie atribútov používateľa bez odhalenia jeho identity. [19]

## Certifikát a štandard X.509

Certifikačné authority vydávajú certifikáty, ktoré potvrdzujú, že daný verejný kľúč je pod správou uvedeného vlastníka (viz obr. 3.1). Vlastník môže byť definovaný presným menom alebo alternatívnym označením, ako je napríklad e-mailová adresa alebo DNS záznam. Organizácie môžu vydať svoje vlastné dôveryhodné koreňové certifikáty, ktoré platia iba v rámci podniku. Webové prehliadače majú zväčša predinštalované koreňové certifikáty známych poskytovateľov certifikátov (SSL certifikáty).

Podľa odporúčania ITU-T[9] je certifikát X.509 štandard v kryptografii pre systémy založené na verejnom kľúči pre jednoduché podpisovanie. Tento štandard ďalej definuje formát certifikátov, zoznamy odvolaných certifikátov a metódy platnosti kontroly certifikátov. X.509 používa vzájomné potvrdzovanie kľúčov CA. CA do svojej časti adresára pridá k užívateľským certifikátom ešte certifikáty kľúčov iných CA. [20]



Obr. 3.1: VUT SAML Self-Signing Certificate

## Metadata v SAML

Metadáta sú počítačom spracovateľné štruktúrované informačné objekty popisujúce vlastnosti iných informačných objektov. Ich význam vzrastá s pokrokom a vývojom digitálnych objektov a nástrojov. Pomáhajú nielen pri riadení databáz, pri popíšaní ich štruktúry, vlastností, či umiestnenia, ale užívateľom pomáhajú zlepšovať vyhľadávanie, orientáciu, triedenie či pochopenie samotných služieb. Slúžia nám na dokumentáciu a zálohovanie dát.

Aby sme mohli metadáta spravovať, existujú špecializované nástroje a metadátový softvér. Keďže organizácie si potrebujú metadáta medzi sebou vymieňať alebo zdieľať, je dôležité, aby boli v súlade s normami a štandardmi predpísanými jednotlivými operátormi. Tie zjednocujú štruktúru, sémantiku aj syntax metadát. V prvom rade musí byť tvorba metadát na WWW dôveryhodná a spoľahlivá, čo znamená, že v celkovej architektúre hrajú rolu najmä dokumenty s príslušnou URI, ktoré obsahujú význam a definície vlastností, ktoré sú dané podľa metodiky na základe špecifikácie metadát.

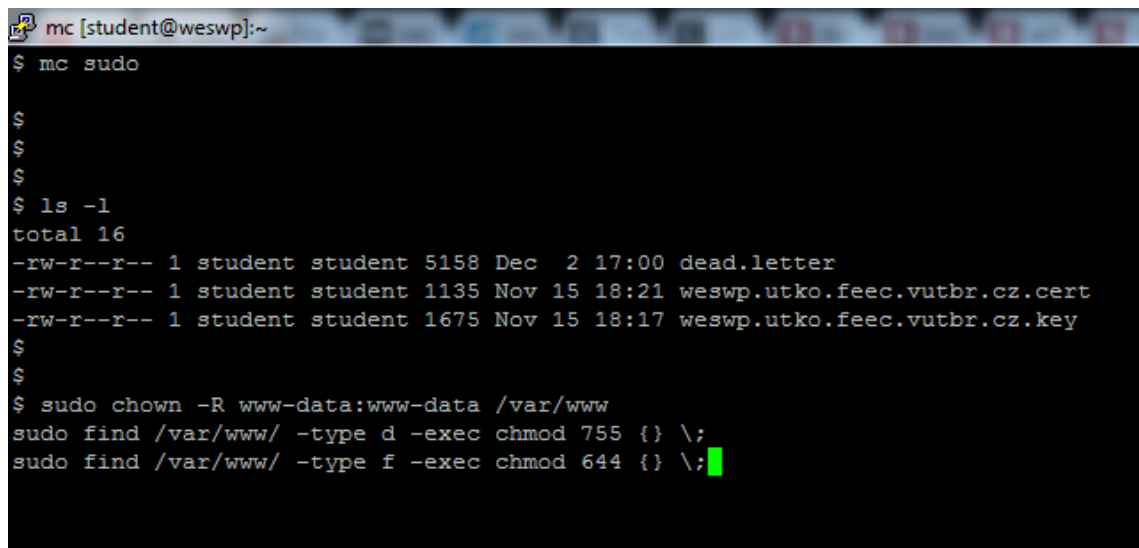
## Shibboleth

Shibboleth je open-source softvér, ktorý ponúka riešenie prepájania užívateľov na dostupné aplikácie zaregistrovaných organizácií. Je postavený na báze protokolu SAML, dátový štandard založený na jazyku XML umožňujúci výmenu autentizačných dát medzi poskytovateľom služby a poskytovateľom identity. Umožňuje autentizáciu a jednotné prihlásenie užívateľov v prostredí webu. V súčasnosti sú prostredníctvom Shibbolethu dostupné napríklad zdroje platformem EBSCOhost, Web Of Knowledge, OvidSP, ebrary, Elsevier Science Direct a Scopus, Cambridge Journals, British Medical Journals a WilsonWeb. [22]

## 4 VÝSLEDKY ŠTUDENTSKEJ PRÁCE

Prvým a najzákladnejším krokom pre realizáciu a splnenie zadania bakalárskej práce bolo vytvorenie elementárnej HTML štruktúry stránky na báze WordPressu. Pred tým, ako mohli byť nainštalované a nastavené základné parametre WordPressu, bolo potrebné sa prihlásiť poskytnutými prihlasovacími údajmi do systému spusteného linuxového servera fungujúceho na VUT v Brne.

Aby mohol server správne fungovať, vybrané komponenty boli nainštalované v rámci samostatného apache servera. Je to program, ktorý je nainštalovaný na hardverovom prístroji s prístupom na internet a stará sa o poskytovanie obsluhy a služieb jednotlivým návštevníkom (zasielanie patričných stránok).



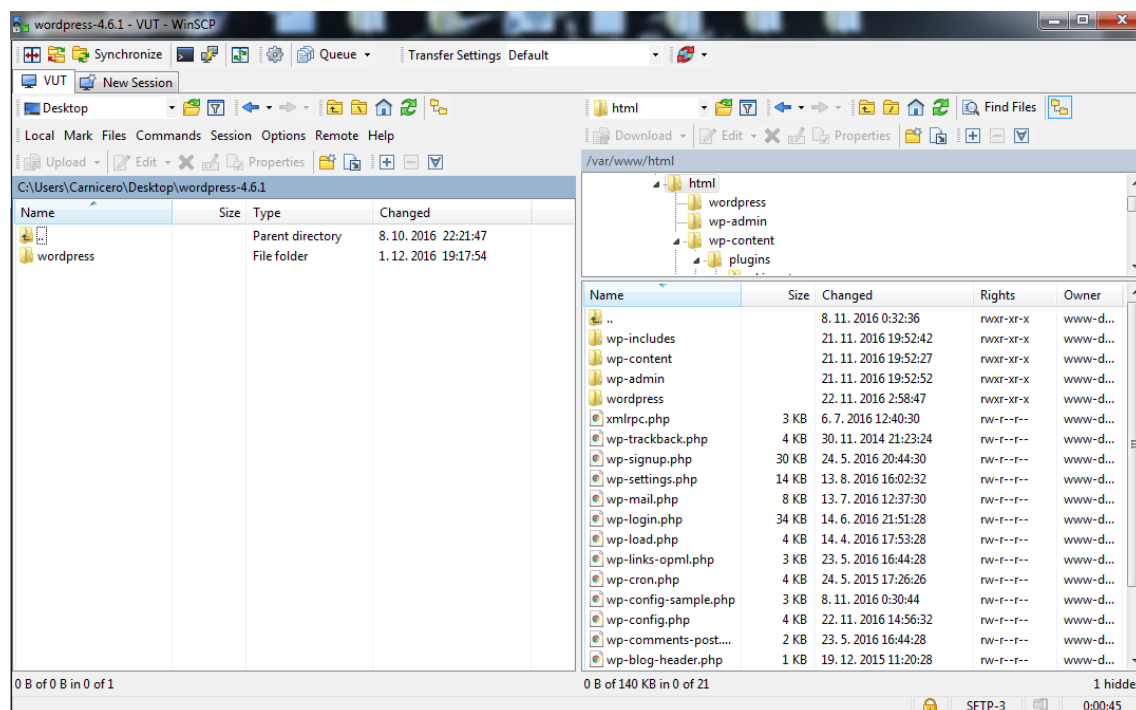
```
mc [student@weswp]:~  
$ mc sudo  
$  
$  
$  
$ ls -l  
total 16  
-rw-r--r-- 1 student student 5158 Dec  2 17:00 dead.letter  
-rw-r--r-- 1 student student 1135 Nov 15 18:21 weswp.utko.feec.vutbr.cz.cert  
-rw-r--r-- 1 student student 1675 Nov 15 18:17 weswp.utko.feec.vutbr.cz.key  
$  
$  
$ sudo chown -R www-data:www-data /var/www  
sudo find /var/www/ -type d -exec chmod 755 {} \;  
sudo find /var/www/ -type f -exec chmod 644 {} \;
```

Obr. 4.1: Prístup k linuxovému serveru

V prvom rade bola zriadená virtuálna linka, tzv. VPN sieťová brána, za účelom pripojenia sa do intranetu na VUT v Brne pomocou šifrovaného SSH spojenia, pre zachovanie bezpečného prenosu dát. Následne bolo potrebné prihlásiť sa vybraným softvérom na poskytnutý server. Prostredníctvom sérií linux CMD príkazov, medzi inými chmod a chown na ovládanie prístupov, bolo zaručené, že sú nastavené práva, teda že sú prístupné všetky z priečinkov a súborov, ktorým bolo potrebné upravovať zdrojové kódy, respektíve v budúcnosti ich pozmeniť. (obr. 4.1)

## 4.1 Inštalácia wordpressu.org

Pre správne fungovanie WordPressu na serveri bola vykonaná inštalácia PHP, vzhľadom na fakt, že samotný WordPress je napísaný v jazyku PHP a MySQL, čo je open-source databázový systém pre ukladanie všetkých druhov informácií v súvislosti s inštaláciou WordPressu, ako sú napr. príspevky, komentáre či nastavenia. Pri mojej práci boli použité verzie Php 5.6 a MySQL verzie 10.5.



Obr. 4.2: Prístup k wordpressu cez FTP

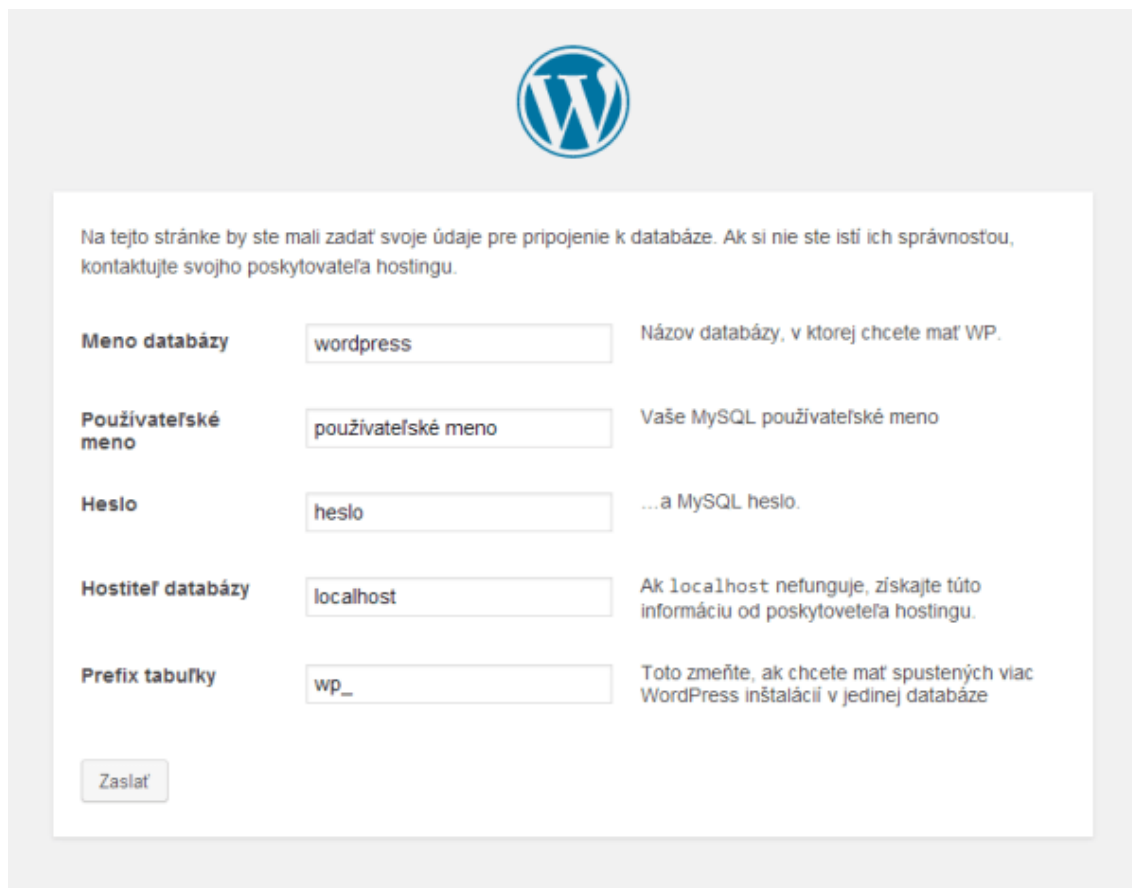
Inštalácia WordPressu je jednoduchý a časovo nenáročný proces. Najprv boli na stránke <https://wordpress.org/> stiahnuté potrebné inštalačné súbory. Obsah priečinka WordPress bol rozbalený a pomocou FTP protokolu nahraný na vzdialený webový server do koreňového adresára.

WordPress pre svoje fungovanie potrebuje databázu MySQL, ale bolo taktiež možné použiť jej novodobú alternatívu v podobe databázy MariaDB.

Ukladanie dát do databáz má množstvo výhod:

1. ak máme text uložený v databáze, vieme rýchlo vyhľadať dáta, ktoré potrebujeme,
2. uľahčenie prístupu pri manipulácii s dátovými súbormi,
3. stránka sa mení zo statickej na dynamickú.

Pre moju prácu bol využitý práve nástroj MySQL. Prihlasovanie do systému bolo realizované cez správcovský systém PhpMy-Admin, ktorým je možné prihlasovať sa a spravovať MySQL databázový systém cez webové rozhranie. Databáza bola vytvorená v administrácii hostingu (viď obr. 2.2), aby bol týmto vyhradený priestor pre ukladanie súborov WordPressu.



Na tejto stránke by ste mali zadať svoje údaje pre pripojenie k databáze. Ak si nie ste istí ich správnosťou, kontaktujte svojho poskytovateľa hostingu.

<b>Meno databázy</b>	<input type="text" value="wordpress"/>	Názov databázy, v ktorej chcete mať WP.
<b>Používateľské meno</b>	<input type="text" value="používateľské meno"/>	Vaše MySQL používateľské meno
<b>Heslo</b>	<input type="text" value="heslo"/>	...a MySQL heslo.
<b>Hostiteľ databázy</b>	<input type="text" value="localhost"/>	Ak localhost nefunguje, získajte túto informáciu od poskytovateľa hostingu.
<b>Prefix tabuľky</b>	<input type="text" value="wp_"/>	Toto zmeňte, ak chcete mať spustených viac WordPress inštalácií v jednej databáze

Obr. 4.3: Vytváranie databázy

Po vytvorení databázy pomocou údajov vytvoreného používateľa a názvu databázy bol WordPress s databázou prepojený a presmerovaný vyplnením údajov a využitím hesla pre bezpečnostný kľúč.

Pre pokračovanie inštalácie bolo treba presunúť sa do zdrojového priečinka WordPressu vytvoriť súbor wp-config.php, alternatívne bola postačujúca úprava implementovaného súboru wp-config-sample.php, prostredníctvom ktorého vieme komunikovať s databázou, upravovať nastavenia či získať dáta uložené v databáze. Postupnými krokmi bolo treba vyplniť požadujúce údaje a dokončiť inštaláciu.

Ďalším krokom v mojej práci bolo vybratie vhodnej témy, resp. šablóny. Bolo potrebné upraviť základné nastavenia, prispôbiť vzhľad požiadavkám zadania mo-

difikáciou zdrojového style.css súboru a vyhľadať vhodné nástroje pre efektívne fungovanie stránky. Vybratím najvhodnejšieho prostriedku na vytvorenie podmienok pre vygenerovanie metadát a následným implementovaním týchto technologických prvkov do systému WordPressu bolo vytvorené vhodné prostredie pre realizáciu výmeny metadát s poskytovateľom identít.

## 4.2 SAML a návrh realizácie prihlasovania

### 4.2.1 Self-signed certificate

Existujú mnohé certifikačné authority, ktoré poskytujú X.509 certifikáty bez poplatku na TLS šifrovanie slúžiacich na zabezpečenie webových stránok. Taktiež je možné vytvorenie a vygenerovanie certifikátu priamo na serveri, teda priamo v systéme, a zaradenie do WordPressu, čo je spôsob, akým bol certifikát vygenerovaný v mojej práci. Táto metóda sa osvedčila ako najprijateľnejšia, keďže na rozdiel od viacerých sprostredkovateľov, ktorý službu vygenerovania certifikátu poskytujú, je vylúčené, aby prístup k nemu mala tretia strana a teda by bola ohrozená bezpečnosť.

Najdôležitejším bodom pri vkladaní certifikátu a vygenerovaného kľúča je, aby nedošlo k modifikácii alebo k zámene kľúčov, keďže museli byť zhodné. V prípade IDP certifikátu bola nutná transformácia na tzv. „certificate fingerprint“, aby bol zachovaný správny formát a certifikát bol akceptovaný.

### 4.2.2 Výmena metadát

Na univerzite VUT v Brne je použitá aplikácia SimpleSAMLphp na umožnenie jednotného prihlasovania podľa štandardu SAML 2.0. Univerzita si tak v podobe aplikácie zabezpečuje členstvo a fungovanie v rámci federácie eduID.cz.

#### Výhody SSO:

- sprístupnenie služieb pomocou jedného prihlásenia,
- v niektorých prípadoch nie je nutné poznať užívateľské heslo, keďže aplikácia dôveruje údajom od poskytovateľa identít,
- ak sa využívajú dostatočne silné heslá, je menšie riziko prezradenia hesla.



## Nevýhody SSO :

- predstavuje bezpečnostné riziko, ak klienti využívajú slabé heslá alebo je ich bezpečnosť znížená zo strany útočníka škodlivým softwérom, v tom prípade sú ohrozené prístupy do všetkých systémov,
- informácie o prihlásení sú sprístupnené strane poskytovateľa identít,
- je nutnosť získať SSO ako samostatný produkt, čo zatažuje rozpočty firiem.

Výpis 4.1: Ukážka metadát SP

```
<md:EntityDescriptor xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata"
    validUntil="2017-05-25T12:00:55Z"
    cacheDuration="PT604800S"
    entityID="https://www.PrikladSP.com">
  <md:SPSSODescriptor AuthnRequestsSigned="false" WantAssertionsSigned="
    false" protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:
    protocol">
    <md:SingleLogoutService Binding="urn:oasis:names:tc:SAML:2.0:
      bindings:HTTP-Redirect"
        Location="https://www.PriklaSP.com" />
    <md:NameIDFormat>urn:oasis:names:tc:SAML:1.1:nameid-format:
      unspecified</md:NameIDFormat>
    <md:AssertionConsumerService Binding="urn:oasis:names:tc:SAML:2.0:
      bindings:HTTP-POST"
        Location="SP"
        index="1" />
  </md:SPSSODescriptor>
  <md:ContactPerson contactType="technical">
    <md:GivenName>Administrator</md:GivenName>
    <md:EmailAddress>PrikladSP@prikklad.com</md:EmailAddress>
  </md:ContactPerson>
</md:EntityDescriptor>
```

Keď dostane IDP od SP podnet, skontroluje databázu s informáciami, ktorej dôveruje, overí si metadáta a pošle odpoveď. Rovnakým spôsobom pomocou týchto metadát IDP vie rozlíšiť, kam má presmerovať používateľa po úspešnej autentifikácii. SP overí podpis potvrdenia pomocou verejného kľúča IDP, a tak SP vie, že IDP nie je oklamáný a môže fungovať medzi nimi výmená informácií na báze dôvery. [21]

#### Výpis 4.2: Ukážka metadát IDP

```
<md:EntityDescriptor xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata"
    validUntil="2017-05-25T10:57:21Z"
    cacheDuration="PT1496141841S"
    entityID="www.PrikladorganizacieIDP.com">
  <md:IDPSSODescriptor WantAuthnRequestsSigned="false"
    protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
    <md:SingleLogoutService Binding="urn:oasis:names:tc:SAML:2.0:
      bindings:HTTP-Redirect"
        Location="https://www.PrikladorganizacieIDP
          .com"/>
    <md:NameIDFormat>urn:oasis:names:tc:SAML:1.1:nameid-format:
      unspecified</md:NameIDFormat>
    <md:SingleSignOnService Binding="urn:oasis:names:tc:SAML:2.0:
      bindings:HTTP-Redirect"
        Location="IDP"/>
  </md:IDPSSODescriptor>
  <md:ContactPerson contactType="technical">
    <md:GivenName>Administrator</md:GivenName>
    <md:EmailAddress>Prikladorganizacie@priklad.com</md:EmailAddress>
  </md:ContactPerson>
</md:EntityDescriptor>
```

K aplikácii sa vieme dostať na adrese <https://www.vutbr.cz/SSO>, konkrétne :

- ako poskytovateľ identít : <https://www.vutbr.cz/SSO/saml2/idp>
- ako poskytovateľ služieb : <https://www.vutbr.cz/SSO/saml2/sp>

V momente, ako boli k dispozícii metadáta SP, viď obr. 4.1, bolo potrebné požiadať operátora o zaradenie metadát do centrálnej databázy. IDP metadáta spracuje a zašle metadáta danej organizácii, viď na obr. 4.2. Po úspešnej výmene metadát boli vložené požadujúce atribúty do aplikácie mojeID.

#### 4.2.3 Nastavenie atribútov

Atribúty v štandarde SAML sú textové reťazce s unikátnym označením, použité ako identifikátory vydané konkrétnym poskytovateľom identít. Vďaka týmto atribútom prebieha proces autentizácie, a teda SAML vie rozlíšiť, o ktorý parameter sa jedná (prihlasovacie meno, heslo, e-mail). Táto služba sa nazývaná mojeID. Poskytovateľia služieb zapojení do federácie eduID.cz majú k dispozícii túto službu aj napriek tomu, že nie sú priami užívatelia nástroja mojeID. Služba eduID.cz-mojeID Bridge v ďalšom kroku vykoná preklad medzi mojeID a eduID. S nastavením týchto bezpečnostných prvkov a zaradením metadát do systému bolo prihlásenie cez VUT login kompletne.

## 4.3 Návrh podporovaných aplikácií

### 4.3.1 Fórum

Po prihlásení sa v menu v sekcii „Studium“ zobrazí kolonka „Forum“. Toto fórum funguje na princípe webovej PHP aplikácie, podľa zadania bakalárskej práce. Po vytvorení fóra administrátorom je možné pridávať jednotlivé príspevky, či už do novej témy alebo už vytvorenej. Navyše je v aplikácii vedená štatistika počtu všetkých prihlásených, či zaregistrovaných ľudí.

### 4.3.2 Odber noviniek

Podpora aplikácie odberu noviniek je umiestnená na pravom paneli vytvoreného portálu, konfigurovaná ako tzv. widget, kde záujemca, ktorý chce vedieť o zmene, či už novom príspevku, osobitej správe vybranej skupine či správe Cron vyplní svoje údaje (meno a e-mail) a je zaradený do zoznamu príjemcov. Administrátor má možnosť užívateľa zaradiť do spomínaných skupinových správ, osobitne kontaktovať či úplne vymazať.

Kedže WordPress v základnej konfigurácii nepodporuje zasielanie e-mailových správ, resp. protokol potrebný pre tento úkon, bolo potrebné docieľiť aby WordPress komunikoval s vhodným protokolom a e-mailovým serverom.

#### Nastavovanie SMTP protokolu

Pre účely zasielania automatických e-mailov pri aktualizovaní noviniek bol vytvorený dočasný účet prostredníctvom služby Gmail (weswp.utko.feec.vutbr.cz@gmail.com). Ako protokol na podporu e-mailových správ bol vybraný SMTP protokol a Gmail SMTP server pre zasielanie správ z Gmail účtu pomocou e-mailového softvéru.

Nastavenie softvéru Postman SMTP:

1. vytvorenie e-mailovej schránky a stanovenie tejto e-mail adresy ako východiskovej, odkiaľ budú e-maily generované,
2. zvolenie e-mail serveru alebo agenta odosielacej pošty, v tomto prípade zvolený server: smtp.gmail.com,
3. špecifikácia vytvoreného spojenia s e-mail serverom, pre tento prípad socket SMTP-gmail.com na porte 587 s autorizačným protokolom OAuth 2.0,
4. pre vyššiu bezpečnosť a identifikáciu e-mailovým serverom je potrebné zadať vygenerované klientské ID a tajný kľúč využívanej webovej API.

Po overení vyššie uvedených nastavení je server schopný prijímať a využívať zasielanie správ záujemcom pri pridaní nového príspevku.

## Webová API

Web API možno jednoducho nazvať webovou službou. V kontexte vývoja webu API využíva správu štruktúry odpovede vo väčšine prípadov vo formáte JSON alebo XML. Či už na platforme Google alebo v inom prostredí sa využívajú práve pri komunikácii s autorizačným serverom. Pre účely tejto práce klient, vytvorená aplikácia, nedostane priamo prihlasovacie údaje, ale prihlásenie prebieha na stránke, na ktorej bol vytvorený účet. Obdobným spôsobom je následne vytvorená aj webová API (tomto prípade na platforme Google).

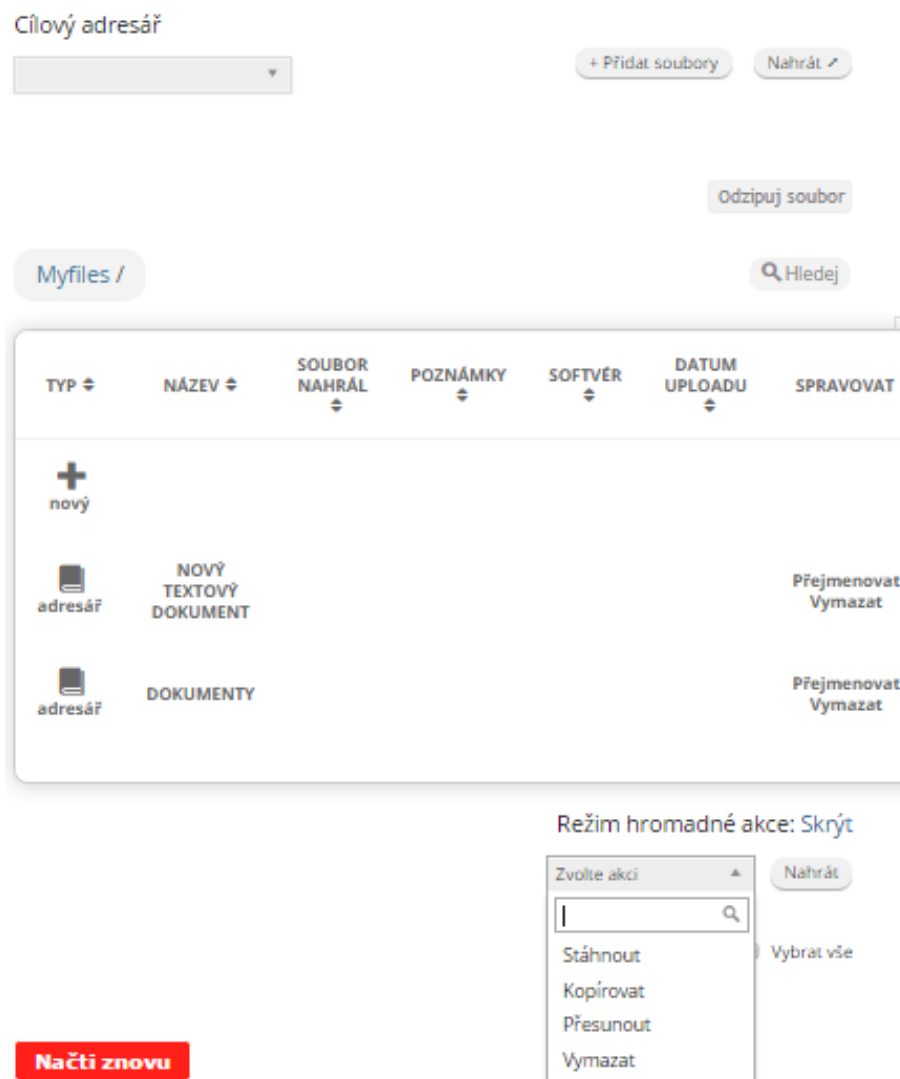
Musí tomu ale predchádzať registrácia klienta na autorizačnom serveri, pričom pri registrácii sa vygenerujú tzv. Client secret a Client Id, unikátne identifikátory, ktoré sa následne používajú ako overovací prostriedok pri komunikácii medzi klientom a autorizačným serverom.

### 4.3.3 Úložisko

Po prihlásení je na stránke užívateľovi prístupná sekcia „Uložisko“ (viz obr. 4.4). Z dôvodu väčšej prehľadnosti je vytvorená stromová štruktúra a umožnená voľba pridávať súbory do vybraných, prípadne taktiež povolenou funkciou „nový“ vytvorených priečinkov. Úložisko nie je limitované typom či formátom súboru, typ súboru sa zobrazuje v kolonke typ s príslušnou ikonou súboru.

Pre nahranie súboru v prom rade zvolíme cestu v sekcii „Cílový adresár“, miesto, kde má byť súbor uložený. Následným kliknutím na tlačítko „+ Pridat soubory“ súbor nahráme do systému. Ukáže sa nám okno s názvom, ktorý môžeme pozmeniť. Ak nechceme nič ďalej meniť, zvolíme „Nahrát“ a súbor je v databáze. Po opätovnom načítaní stránky ho môžeme vidieť v našej tabuľke nahraných súborov. Vytvorená tabuľka po nahraní eviduje taktiež užívateľa, v sekcii „Soubor nahrál“ a dátum nahrania súboru v stĺpci „Datum uploadu“.

Pre jednoduché vloženie informácií o súbore nám slúži funkcia „Přejmenovat“, pomocou ktorej v danom riadku pozmeníme prípadne doplníme informáciu o súbore. Pre prípad rýchleho vyhľadávania v databáze je poskytnutá možnosť „Hledej“. Pre odzipovanie súborov nám slúži samostatné tlačítko „odzipovať súbor“. Možnosťou „Vyber soubor“ vybereme .zip súbor a nahráme ho do tzv. „kontajneru“, kde súbor zotrvá dočasne. Tlačítkom „Nahrát“ ho potvrdíme a odzipovaný ho uložíme do databázy.



Obr. 4.4: Návrh webového úložiska

Ak sa rozhodneme pre vymazanie súboru, ktorý viac nepotrebujeme, pod funkciou „Přejmenovat“ sa nachádza funkcia „Vymazat“, ktorá splní tento účel. V administrátorskej sekcii máme k dispozícii taktiež možnosť presunúť súbor funkciou „Přesunout“ alebo kopírovať voľbou „Kopírovat“ v sekcii pri rozbalení rozšírenej voľby režimu „Režim hromadné akce“ na „Zobrazit“.

Akonáhle označíme súbor, s ktorým chceme pracovať, zvolíme akciu, ktorú chceme vykonať, objaví sa na ľavej strane opäť položka „Cílový adresář“, aby aplikácia vedela, do ktorej zložky, prípadne adresára, presúvame alebo kopírujeme daný súbor. Po zvolení akcie odsúhlasíme výber tlačítkom „Nahrát“.

## 4.4 Programovanie modulov a aplikácií

### 4.4.1 Function.php

Function.php (príloha A.1) je koreňovým súborom, kde je možné spravovať všetky prebiehajúce procesy, ktoré majú vplyv na samotné fungovanie prostredia WordPress. Tento krátky skript bolo nutné vytvoriť na spravovanie užívateľov, konkrétne na pevné stanovenie administrátorskej role. Administrátorske práva sú priradené na základe pričleneného ID, ktoré dostane automaticky každý, kto prejde procesom úspešného prihlásenia.

### 4.4.2 Wp-config.php

Pred dokončením inštalácie WordPressu bolo nutné súbor wp-config.php (príloha A.2) upraviť, nastaviť správne atribúty a šifrovacie kľúče pre správnu komunikáciu WordPressu s MySQL.

### 4.4.3 Style.css

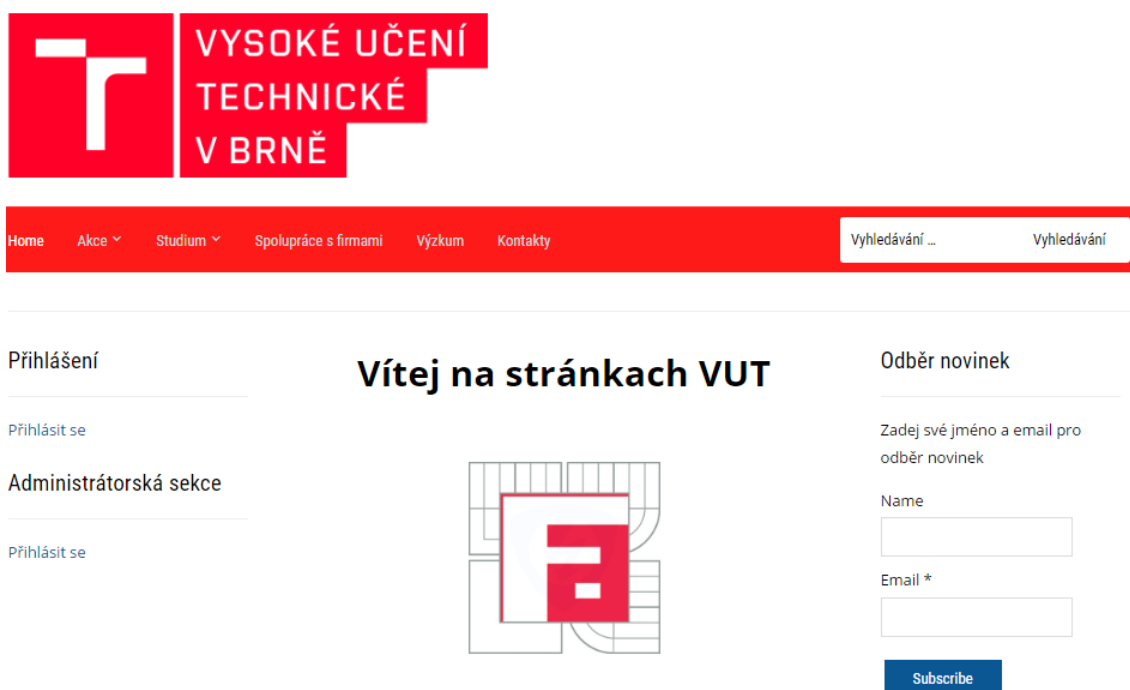
Style.css (príloha A.3) je zdrojovým úložiskom kaskádových štýlov. Pre úpravu vzhľadu do súčasnej podoby bolo nutné vykonať viacere zmeny v jazyku CSS.

### 4.4.4 Unzip.php

Unzip.php (príloha A.4) je PHP skript vytvorený na to, aby splňal funkciu odzipovania súboru a následného extrahovania. Po stlačení tlačítka sa nám objaví stránka na základoch jazyka HTML. Ak sa rozhodneme nahráť súbor, aktivujeme skript. V tomto prípade sa jedná o PHP metódu POST, ktorá sa hodí na odosielenie väčšieho množstva dát (oproti jednoduchšej PHP metóde GET). Skript funguje na princípe formulára, ktorého údaje sú prenášané v tele HTTP požiadavku.

### 4.4.5 Class.fileup.php

Class.fileup.php (príloha A.4) je skript zodpovedný za nahrávanie súborov bez limitovania formátom či typom súborov v úložisku. Je tvorený sériou formulárov funkcie GET pri prenášaní menších formulárov na konci URL pri spolupráci s obslužným skriptom.



Obr. 4.5: VUT stránka

## 5 ZÁVER

Cielom tejto bakalárskej práce bolo vytvoriť a navrhnúť riešenie pre webový portál na edukatívne účely pomocou systému WordPress pre prihlásenie VUT Logi-nom. V prvej časti bol popísaný úvod do problematiky vytvárania stránok pomocou redakčno-publikačného systému WordPress. Aké su jeho možnosti, jeho verzie, na čo sú využívané, s čím musí užívateľ počítať v prípade, že sa rozhodne pre jednu z nich. WordPress sa ukázal ako veľmi šikovný nástroj pre budovanie solídnej webovej stránky s množstvom možností.

V ďalšej časti boli opisované webové technológie a aplikácie, tzv. „Triáda“, trojica aplikácií, ktoré na seba naväzujú a sú nevyhnutné pri vytváraní dynamickej webovej stránky, ktorá bola podkladom pre túto prácu. Taktiež bol popísaný programovací jazyk PHP a ako sa stal stálym štandardom pri dynamických stránkach, porovnaním HTML a PHP a ich spracovaním požiadavky medzi prehliadačom a serverom. Následne bolo písane o dostupnosti a nevyhnutnosti Apache servera a jeho kompatibilitate s rôznymi operačnými systémami, MySQL a jeho jednoznačná suverénita medzi databázovými systémami, ktorý sa nakoniec ukázal ako veľmi prehľadnou a dobrou alternatívou pri databázach, v neposlednom rade aj pre jeho spoluprácu s programovacím jazykom PHP.

V ďalšej kapitole bol vysvetlený systém jednotného prístupu SSO ako riešenia nútených zmien v rámci bezpečnosti internetu. V tejto kapitole boli taktiež zahrnuté známe štandardy a výhody mnou používaného štandardu SAML, výhody vyžívannej federácie euidid.cz ako riešenia a zvýhodnenia v oblasti bezpečného prenosu informácií na vzajomnej dôvere poskytovateľa identity a poskytovateľa služieb. Sú zhrnuté jej funkcie, výhody, nevýhody a jednotlivé implementácie.

Následne sú popísané jednotlivé služby a technológie, ktoré na ten účel môžu byť použité, a v nasledujúcej časti tejto práce som konkretizoval práve tie, ktoré som využil pre sfunkčnenie bezpečného prihlásenia sa cez stránky VUT do mnou vytvorenej webovej internetovej stránky.

V záverečnej časti som sa venoval návrhom riešení webových aplikácií podporujúcich fotogalérie, odber novinek, diskusné fórum, kalendár akcií a databázu študijných materiálov. Rozoberal som funkcie a nástroje, ktoré som naviazal na úložisko, nahrávanie celých zložiek súborov, v podobe formátu .zip, tak ako aj hromadné sťahovanie zložiek, vpisovanie informácií pre jednotlivé súbory, tak isto ako manipuláciu s nahranými adresármi.

Zadanie bakalárskej práce stanovené na sprístupnenie verejnej a neverejnej sekcie študentom vybraného oboru Audio inžinierstva nebolo možné uskutočniť pre nedostupnosť potrebného modulu na jeho realizáciu. Do dátumu odovzdania bakalárskej



práce nebol Centrom informačných a výpočtových služieb VUT vytvorený skript pre overenie vzťahu k fakulte a oboru prihláseného užívateľa. Odhliadnúc od tohto problému, prihlásenie pomocou VUT loginu a hesla funguje bezchybne. Pre demonštrovanie jeho funkčnosti bolo prihlásenie cez SSO nastavené na dostupný atribút, ktorý rozlišuje študentov na základe príslušnosti k fakulte FEKT, študentom iným fakúlt prihlásenie nie je povolené.

# LITERATÚRA

- [1] WordPress. *SiteGround* [online]. 2004 [cit. 2016-12-12]. Dostupné z: <<https://www.siteground.com/wordpress.htm>>
- [2] About WordPress. *Wordpress.org* [online]. [cit. 2016-12-12]. Dostupné z: <<https://wordpress.org/about/>>
- [3] MCNULTY, Scott. *WordPress: efektivní publikování na webu*. Brno: Zoner Press, 2009. Encyklopedie webdesignera. ISBN 9788074130427. Dostupné z URL: <<http://www.beletrie.eu/data/attachments/WordPress%20-%20efektivn%C3%AD%20publikov%C3%A1n%C3%AD%20na%20webu.pdf>>.
- [4] KOSEK, Jiří. *PHP-tvorba interaktivních internetových aplikací: podrobný průvodce* [online]. Praha: Grada, 1998 [cit. 2016-11-22]. Průvodce (Grada). ISBN 8071693731. Dostupné z URL: <<http://www.kosek.cz/php/php-tvorba-interaktivnich-internetovych-aplikaci.pdf>>.
- [5] 1. díl - Úvod do PHP a webových aplikací. *Itnetwork* [online]. [cit. 2017-05-10]. Dostupné z: <<https://www.itnetwork.cz/php/\zaklady/\php-tutorial-\uvod-do-\webovych-\aplikaci>>
- [6] PHP /základy/. *Tvorba-webu* [online]. 2003 [cit. 2016-12-12]. Dostupné z: <<http://www.tvorba-webu.cz/php/>>
- [7] JANOVSKEÝ, Dušan. Možnosti PHP. *jakpsatweb.cz* [online]. 2005 [cit. 2016-12-12]. Dostupné z: <<https://www.jakpsatweb.cz/php/moznosti-php.html>>
- [8] Apache Server. *Adaptic* [online]. 2005 [cit. 2016-12-12]. Dostupné z: <<http://www.adaptic.cz/znalosti/slovnicek/apache-server/>>
- [9] JANOVSKEÝ, Dušan. PHP – Jak začít. *Jakpsatweb* [online]. [cit. 2016-12-12]. Dostupné z: <<https://www.jakpsatweb.cz/php/jak-zacit.html>>
- [10] PHP a MySQL - Prepojenie 1. časť. *Http://programujte.com/* [online]. [cit. 2016-12-10]. Dostupné z: <<http://programujte.com/clanek/\2006041902\php-a-mysql-prepojenie-1-cast/>>
- [11] TSCHABITSCHER, Heinz. *Gmail SMTP Settings for Sending Mail* [online]. [cit. 2017-05-11]. Dostupné z: <<https://www.lifewire.com/what-are-the-gmail-smtp-settings-1170854>>
- [12] *User Authentication with OAuth 2.0* [online]. , 1 [cit. 2017-05-12]. Dostupné z: <<https://oauth.net/articles/authentication/>>

- [13] NOVÁKOV, Ivan. Web Single Sign On Systems. In: *Cesnet* [online]. 2006 [cit. 2016-12-12]. Dostupné z: <<http://archiv.cesnet.cz/doc/techzpravy/2006/web-sso/>>
- [14] Understanding Enterprise Single Sign-On. *Microsoft Host Integration Server* [online]. 2005 [cit. 2016-12-12]. Dostupné z: <[https://msdn.microsoft.com/en-us/library/aa745042\(v=bts.10\).aspx](https://msdn.microsoft.com/en-us/library/aa745042(v=bts.10).aspx)>
- [15] Definition of Web Single Sign-On (WebSSO). *Hitachi ID Systems* [online]. 2016 [cit. 2016-12-12]. Dostupné z: <<http://hitachi-id.com/concepts/websso.html>>
- [16] BOUŠKA, Petr. Kerberos část 3 - Single Sign-On a protokol Kerberos. In: *Samuraj-cz.com* [online]. 2014 [cit. 2016-12-12]. Dostupné z: <<http://www.samuraj-cz.com/clanek/kerberos-protokol-a-single-sign-on/>>
- [17] Česká akademická federace identit eduID.cz. *Česká akademická federace identit eduID.cz* [online]. [cit. 2016-12-12]. Dostupné z: <<https://www.eduid.cz/cs/index/>>
- [18] MALÝ, Martin. OpenID: Identity, aliasy a vlastní poskytovatel. *Zdrojak* [online]. 2009 [cit. 2016-12-12]. Dostupné z: <<https://www.zdrojak.cz/clanky/openid-identity-aliasy-a-vlastni-poskytovatel/>>
- [19] SAML: How It Works. *Pingidentity* [online]. [cit. 2016-12-12]. Dostupné z: <<https://www.pingidentity.com/en/resources/articles/saml.html>>
- [20] Certifikáty veřejných klíčů. *Webserver.ics.muni* [online]. [cit. 2016-12-12]. Dostupné z: <<http://webserver.ics.muni.cz/bulletin/articles/181.html>>
- [21] BRATKOVÁ, Eva. Metadata jako nový nástroj pro komunikaci webovských informačních zdrojů. *Full.nkp* [online]. Praha, 1999 [cit. 2016-12-12]. Dostupné z: <http://full.nkp.cz/nkk/Nkk9904/9904178.html>  
<<https://www.pingidentity.com/en/resources/articles/saml.html>>
- [22] What's Shibboleth? *Shibboleth* [online]. [cit. 2016-12-12]. Dostupné z: <<https://shibboleth.net/about/>>

## ZOZNAM SYMBOLOV, VELIČÍN A SKRATIEK

API	Application programming interface
AES	Advanced Encryption Standard
CESNET	Czech Education and Scientific NETwork
CGI	Common Gateway Interface
CMD	database management system
CMS	Content Mangaement System
DBMS	database management system
DES	Data Encryption Standard
EDUID	The Education system identifier
FTP	File Transfer Protocol
GPL	General Public License
HTML	HyperText Transfer Protocol
IDP	identity provider
IMAP	Internet Message Access Protocol
IP	Interne Protocol
MySQL	My Structured Query Language
PHP	Hypertext Preprocessor
POP	Post Office Protocol
RC4	Rivest Cipher 4
SAML	Security Assertion Markup Language
SMTP	Simple Mail Transfer Protocol
SQL	Structured Query Language
SOAP	Simple Object Access Protocol
SP	Service provider
SSH	Secure Shell
SSJS	Server Side Javascript
SSO	Single sign-on
URI	Uniform Resource Identifier
VPN	Virtual Private Network
WWW	Word-Wide Web

# ZOZNAM PRÍLOH

<b>A</b>	<b>Zdrojový kód aplikácie</b>	<b>45</b>
A.1	Wordpress súbory . . . . .	45
A.1.1	function.php . . . . .	45
A.1.2	wp-config.php . . . . .	45
A.1.3	style.css . . . . .	46
A.2	Subory webový aplikácií . . . . .	48
A.2.1	unzip.php . . . . .	48
A.2.2	class.fileup.php . . . . .	49
<b>B</b>	<b>Prístup na webovú stránku</b>	<b>54</b>
B.1	Povolenia na prihlásenie: . . . . .	54
B.2	Ako sa prihlásiť: . . . . .	54
B.3	Administrátorský prístup: . . . . .	54
B.4	Dočasne vytvorená emailová schránka . . . . .	54
<b>C</b>	<b>Obsah přiloženého CD</b>	<b>55</b>

# A ZDROJOVÝ KÓD APLIKÁCIE

## A.1 Wordpress súbory

### A.1.1 function.php

Výpis A.1: Function.php

```
1 <?php
2 $user_id = 5;
3     $new_role = 'administrator';
4
5     $result = wp_update_user(array('ID'=>$user_id, 'role'=>$new_role));
6
7     if ( is_wp_error( $result ) ) {
8         // ak by nebol skript úspešný, pravdepodobne účet neexistuje.
9     } else {
10         // účet aktualizovaný
11     }
```

### A.1.2 wp-config.php

Výpis A.2: wp-config.php

```
1 <?php
2 define('DB_NAME', 'wordpress');
3 define('DB_USER', 'root');
4 define('DB_PASSWORD', 'WordPressab9dad0055');
5 define('DB_HOST', 'localhost');
6 define('DB_CHARSET', 'utf8mb4');
7 define('DB_COLLATE', '');
8 define('FS_METHOD', 'direct');
9
10 define('AUTH_KEY',                'NRrLUSp#S`n'6%|B'mzt$NuI-suu0xL3D?mqYj4.Tj0i2t0
    _XgIlrJ$K7t:-$yW2');
11 define('SECURE_AUTH_KEY',        'ji%WdH}dt$@k#{W(4qIG4spIDaX/^NAB*YaS?P?i+NG=q{
    bwf<@EqhT.(DDlW_Su');
12 define('LOGGED_IN_KEY',          '6@:~*S]u_y3ob3h$sUOp7~4~_WB$t}Z+*EKoN/MeFN~X=
    U6HyP>WS2BxjRZ_NdrC');
13 define('NONCE_KEY',              '5+VDQ>f_GPQJ:M_QL=6nN8WXv4g?.ILZM('mH#[PTOQ~
    wKF1$b7#Wsdg|,VO[[B');
14 define('AUTH_SALT',              'BST)wFt0sd}.?:@_ '~1HZ}x[Gz2o}2m8!Kb*py)mh]+|v]#
    pimaA&+f28)0~C?c=|');
15 define('SECURE_AUTH_SALT',       'mEH7~+fF1|Zb|6jpD1?iySczH1gNuoYiuVot@wYpol;
    m4Rv7HNed1(XNGy'r_@<|');
16 define('LOGGED_IN_SALT',         'W3[ckzJY#,1@W_7jleb}2^Fv{{H0oQ1Cg4-qun)z1A!
    EPUNic}R[Imb[t7;xiD|(');
```

```

17 define('NONCE_SALT',          '<4_/e=01Q6|?Y]mx9:r%S]uRcCl7E10MrmZGga'[(ND>.+0
    e'IX.%'IX>b[okG)B');
18
19 $table_prefix  = 'wp_';
20
21 define('WP_DEBUG', false);
22 define('WP_MEMORY_LIMIT', '256M');
23
24 if ( !defined('ABSPATH') )
25     define('ABSPATH', dirname(__FILE__) . '/');
26
27 require_once(ABSPATH . 'wp-settings.php');

```

### A.1.3 style.css

Výpis A.3: style.css

```

1 #search {
2     float: right;
3     background-color: white;
4     padding: 3px 10px 3px 0;
5     margin: 0 0 10px;
6     border-radius: 3px;
7     border-color: red;
8 }
9 #search input {
10     background: none;
11     color: BLACK;
12     border: none;
13     text-indent: 10px;
14 }
15 .menu {
16     font-size: 0;
17     color: black;
18 }
19 #main-nav ul ul li a {
20     padding: 5px 10px;
21 }
22 #main-nav {
23     margin: 30px 0 0;
24     padding: 10px 0 0;
25     position: relative;
26     line-height: 1.8;
27     z-index: 300;
28     display: inline-block;
29     width: 100%;
30     border-top: 1px solid rgba(255,255,255,0.1);
31     color: blue;
32     background: #ff1a1a;
33 }

```

```

34 .main-navbar {
35     display: inline-block;
36     vertical-align: middle;
37     margin-top: 5px;
38 .navbar-nav ul {
39     font-size: 14px;
40     left: -15px;
41     width: 220px;
42     margin-top: 0;
43     background-color: #0A5794;
44 }
45 .navbar-nav ul a {
46     padding: 10px 15px;
47     border-bottom: 1px solid #11609E;
48     padding: 5px 15px;
49     font-size: 16px;
50 }
51 .uvitanie {
52     font-size: 38px;
53     font-weight: bold;
54     text-align: center;
55 }
56 html {
57     color: black;
58     font-family: "OpenSans", sans-serif;
59     font-size: 100%;
60     line-height: 1;
61     overflow-y: scroll;
62     text-align: left;
63     -webkit-text-size-adjust: 100%;
64     -ms-text-size-adjust: 100%;
65     -webkit-font-smoothing: antialiased;
66     -moz-osx-font-smoothing: grayscale;
67 }
68 body {
69     font-size: 16px;
70     margin: 0;
71     padding: 0;
72     background-color: #fff;
73 }

```



## A.2 Subory webový aplikací

### A.2.1 unzip.php

Výpis A.4: Unzip.php

```
1 <?php
2 function rmdir_recursive($dir) {
3     foreach(scandir($dir) as $file) {
4         if ('.' === $file || '..' === $file) continue;
5         if (is_dir("$dir/$file")) rmdir_recursive("$dir/$file");
6         else unlink("$dir/$file");
7     }
8     rmdir($dir);
9 }
10 if($_FILES["zip_file"]["name"]) {
11     $filename = $_FILES["zip_file"]["name"];
12     $source = $_FILES["zip_file"]["tmp_name"];
13     $type = $_FILES["zip_file"]["type"];
14     $name = explode(".", $filename);
15     $accepted_types = array('application/zip', 'application/x-zip-compressed',
16         , 'multipart/x-zip', 'application/x-compressed');
17     foreach($accepted_types as $mime_type) {
18         if($mime_type == $type) {
19             $okay = true;
20             break;
21         }
22     }
23     $continue = strtolower($name[1]) == 'zip' ? true : false;
24     if(!$continue) {
25         $message = "Soubor, který se snažíte nahrát není .zip soubor, skuste_
26             znovu.";
27     }
28     $path = dirname(__FILE__).'/';
29     $filenoext = basename ($filename, '.zip');
30     $filenoext = basename ($filenoext, '.ZIP');
31     $targetdir = $path . $filenoext;
32     $targetzip = $path . $filename;
33
34     if (is_dir($targetdir)) rmdir_recursive ( $targetdir);
35     mkdir($targetdir, 0777);
36
37     if(move_uploaded_file($source, $targetzip)) {
38         $zip = new ZipArchive();
39         $x = $zip->open($targetzip);
40         if ($x == true) {
41             $zip->extractTo($targetdir);
42             $zip->close();
43             unlink($targetzip);
44         }
45         $message = "Vas soubor byl odzipovan a ulozen.";
```

```

44 $url='http://wewp.utko.feec.vutbr.cz/index.php/uloziste/';
45 echo '<META_HTTP-EQUIV=REFRESH_CONTENT="1;_'. $url. '>';
46 } else {
47     $message = "Problem_s_uploadom_souboru._Skuste_znovu.";
48     $url='http://wewp.utko.feec.vutbr.cz/index.php/uloziste/';
49     echo '<META_HTTP-EQUIV=REFRESH_CONTENT="1;_'. $url. '>';
50 }
51 }

```

## A.2.2 class.fileup.php

Výpis A.5: class.fileup.php

```

1 <?php
2 if(class_exists('fileaway_attributes') && !class_exists('fileup'))
3 {
4     class fileup extends fileaway_attributes
5     {
6         public function __construct()
7         {
8             parent::__construct();
9             add_shortcode('fileup', array($this, 'sc'));
10        }
11        public function sc($atts)
12        {
13            $uid = rand(0, 9999);
14            $get = new fileaway_definitions;
15            if(isset($atts['style'])) $atts['theme'] = $atts['style']; // legacy
16            if(isset($atts['uploader']) && $atts['uploader'] === 'true') $atts['
                uploader'] = 'name'; // legacy
17            extract($get->pathoptions);
18            extract($this->correct(wp_parse_args($atts, $this->fileup), $this->
                shortcodes['fileup']));
19            if($devices == 'mobile' && !$get->is_mobile) return;
20            elseif($devices == 'desktop' && $get->is_mobile) return;
21            if(!fileaway_utility::visibility($hidefrom, $showto)) return;
22            if($this->op['javascript'] == 'footer') $GLOBALS['
                fileaway_add_scripts'] = true;
23            if($this->op['stylesheet'] == 'footer') $GLOBALS['fileaway_add_styles
                '] = true;
24            // Build Initial Directory
25            $base = $base == 's2member-files' ? fileaway_utility::replacefirst(
                WP_PLUGIN_DIR.'/s2member-files', $chosenpath, '') : $this->op['
                base'].$base];
26            $base = trim($base, '/');
27            $base = trim($base, '/');
28            $sub = $sub ? trim($sub, '/') : false;
29            $dir = $sub ? $base.'/'.$sub : $base;
30            extract(fileaway_utility::dynamicpaths($dir));
31            $dir = str_replace('//', '/', "$dir");

```

```

32     $debugpath = $chosenpath.$dir;
33     $dir = $problemchild ? $install.$dir : $dir;
34     if(!is_dir("$dir") && $mkdir && (!$private_content || (
        $private_content && $logged_in && strpos($dir, 'fa-nullmeta')
        === false)))
35         if(mkdir($rootpath.$dir, 0775, true)) fileaway_utility::indexmulti(
            $rootpath.$dir, $chosenpath);
36     if($private_content && !is_dir("$dir")) return;
37     $start = "$dir";
38     if($matchdrawer)
39     {
40         $fixedlocation = true;
41         $drawerid = $matchdrawer && $matchdrawer !== 'true' ? $matchdrawer
            : null;
42         include fileaway_dir.'/lib/inc/inc.open-drawer.php';
43         $start = "$dir";
44     }
45     $pathparts = explode('/', $start);
46     $basename = end($pathparts);
47     $fixed = $start;
48     $fixed = $fixedlocation ? ($problemchild ? fileaway_utility::
        replacefirst($fixed, $install, '') : $fixed) : null;
49     $path = '<input type="hidden" id="ssfa-upload-actionpath-'. $uid .' "
        value="'. $fixed .' " data-basename="'. $basename .' " data-start="'.
        $start .' " />';
50     // File Type Permissions
51     $types = array();
52     if($filetypes)
53     {
54         $filetypes = preg_split('/(,|,)/', $filetypes, -1,
            PREG_SPLIT_NO_EMPTY);
55         if(is_array($filetypes)) foreach($filetypes as $type) $types[] =
            strtolower(str_replace(array('.', '_'), '', $type));
56     }
57     if($filegroups)
58     {
59         $groups = preg_split('/(,|,)/', strtolower(str_replace('_', '',
            $filegroups)), -1, PREG_SPLIT_NO_EMPTY);
60         foreach($get->filegroups as $group => $discard) if(in_array($group,
            $groups)) $types = array_merge($types, $get->filegroups[$group
            ][2]);
61     }
62     if(count($types) > 0)
63     {
64         $types = array_unique($types);
65         asort($types);
66         $filetypes = '[' . implode(',', $types) . ']';
67     }
68     else $filetypes = false;
69     $permitted = ($filetypes || $filegroups) && $action == 'permit' ?
        $filetypes : 'false';

```

```

70     $prohibited = ($filetypes || $filegroups) && $action == 'prohibit' ?
        $filetypes : 'false';
71     // Configure Settings
72     $name = $name ? $name : "ssfa-meta-container-$uid";
73     $width = is_numeric(preg_replace('[\D]', '', $width)) ? preg_replace(
        '[\D]', '', $width) : '100';
74     $width = "width:$width$perpx;";
75     if($width == '100' && $perpx == '%') $align = 'none';
76     $clearfix = $align == 'none' ? '<div class="ssfa-clearfix"></div>' :
        null;
77     $float = 'float:'. $align.'';
78     $margin = ($width != 'width:100%' ? ($align == 'right' ? 'margin-
        left:15px;' : 'margin-right:15px;') : null);
79     $inlinestyle = $width.$float.$margin;
80     $multiple = $single ? '' : 'multiple=multiple';
81     $addfiles = $single ? __('+AddFile', 'file-away') : __('+AddFiles
        ', 'file-away');
82     $overwrite = $overwrite == 'true' ? 'true' : 'false';
83     $uploadlabel = $uploadlabel ? $uploadlabel : __('FileUp&#10138;', '
        file-away');
84     $pathcheck = $problemchild ? fileaway_utility::replacefirst($start,
        $install, '') : $start;
85     $uploadedby = $uploader ? get_current_user_id() : 0;
86     $uploadtype = $uploader ? $uploader : 'false';
87     // Configure Max File Size Setting
88     $max_file_size = trim(preg_replace('[\D]', '', $maxsize));
89     $max_size_type = trim(strtolower($maxsizetype));
90     $max_file_size = is_numeric($max_file_size) ? $max_file_size : 10;
91     $max_size_type = in_array($max_size_type, array('k','m','g')) ?
        $max_size_type : 'm';
92     $ms = $max_file_size.$max_size_type;
93     $ms = fileaway_utility::ini(false, true, false, $ms);
94     $pms = fileaway_utility::ini('post_max_size');
95     $ums = fileaway_utility::ini('upload_max_filesize');
96     $maxsize = $pms < $ms ? $pms : $ms;
97     $maxsize = $ums < $maxsize ? $ums : $maxsize;
98     // Initialize Settings
99     $fixedsetting = $fixedlocation ? '$fixed.' : 'false';
100    $initialize =
101        '<script>'.
102            'FileUpConfig['.$uid.']={'.
103                'form_id:"ssfa_fileup_form_'.$uid.'",'.
104                'uid:'.$uid.','.
105                'nonce:'.$wp_create_nonce('fileaway-fileup-nonce').','.
106                'container:'.$name.','.
107                'table:'.$theme.','.
108                'iconcolor:'.$iconcolor.','.
109                'maxsize:'.$maxsize.','.
110                'permitted:'.$permitted.','.
111                'prohibited:'.$prohibited.','.
112                'fixed:'.$fixedsetting.','.

```

```

113         'pathcheck:_'.'.$pathcheck.'",_'.
114         'uploader:_'.'.$uploadedby.'",_'.
115         'identby:_'.'.$uploadtype.'",_'.
116         'overwrite:_'.'.$overwrite.'",_'.
117         'loading:_'.'.$fileaway_url.'/lib/img/ajax.gif"_''.
118     '};_'.
119     '</script>';
120     $fadeit = $fadein ? ($fadein == 'opacity' ? 'opacity:0;' : 'display:
        none;') : null;
121     if($fadein)
122     {
123         $fadescript = $fadein == 'opacity' ? '.animate({opacity:"1"},_'.
            $fadetime.');' : '.fadeIn('.$fadetime.');';
124         $initialize .= '<script>jQuery(document).ready(function($) {_
            setTimeout(function(){_$("#div#ssfa_fileup_container_'.'.$uid.'")'
            ._$fadescript.'_'},_1000);_});</script>';
125     }
126     // Form Output
127     if(!is_dir($debugpath)) return current_user_can('administrator')
128         ? __('File Up Admin Notice: The initial directory specified does
            not exist:', 'file-away').<br>.$debugpath
129         : null;
130     $dropdown = $fixedlocation
131         ? null
132         : '<div_id="ssfa-fileup-path-container-'.'.$uid.'"_style="display:
            inline-block;_float:left;">'.
133         '<div_id="ssfa-fileup-directories-select-container-'.'.$uid.'">'.
134         '<label_for="ssfa-fileup-directories-select-'.'.$uid.'"_'.
135         '>style="display:block!important;_margin-bottom:5px!important;
            _text-align:left;">.__('Destination Directory', 'file-
            away').
136         '</label>'.
137         '<select_name="ssfa-fileup-directories-select-'.'.$uid.'"_id="
            ssfa-fileup-directories-select-'.'.$uid.'"_'.
138         '>class="chozed-select_ssfa-fileup-directories-select"_data-
            placeholder="&nbsp;">'.
139         '<option></option>'.
140         '<option_value="'.$start.'">'.$basename.'</option>'.
141         '</select>'.
142         '<br>'.
143         '<div_id="ssfa-fileup-action-path-'.'.$uid.'"_style="margin-top:5
            px;_min-height:25px;">'.
144         '<img_id="ssfa-fileup-action-ajax-loading-'.'.$uid.'"_src="'.
            $fileaway_url.'/lib/img/ajax.gif"_''.
145         '>style="width:15px;_margin:0_0_0_5px!important;_box-shadow:
            none!important;_display:none;">'.
146         '</div>'.
147         '</div>'.
148         '</div>';
149     $form =
150         $clearfix.

```

```

151         '<div_␣id="ssfa_fileup_container_' . $uid . '"_class="
            ssfa_fileup_container_␣' . $class . '"_data-uid="' . $uid . '"_data-mn="
            ' . wp_create_nonce('fileaway-manager-nonce') . '"_style="' .
            $inlinestyle . '␣' . $fadeit . '">' .
152         '<form_␣name="ssfa_fileup_form_' . $uid . '"_id="ssfa_fileup_form_' .
            $uid . '"_action="javascript:void(0);"_enctype="multipart/form-
            data">'
153         . $path . $dropdown .
154         '<div_␣class="ssfa_fileup_buttons_container"_style="text-align:
            right;">' .
155         '<span_␣class="ssfa_fileup_wrapper"_style="text-align:left;">'
            .
156         '<input_␣type="file"_name="ssfa_fileup_files_' . $uid . '[]"_id
            ="ssfa_fileup_files_' . $uid . '"_
157         'class="ssfa_hidden_browse"' . $multiple . '_data-uid="' . $uid
            . '"_/>' .
158         '<span_␣class="ssfa_add_files">' . $addfiles . '</span>' .
159         '<span_␣id="ssfa_submit_upload_' . $uid . '"_data-uid="' . $uid . '
            ">' . $uploadlabel . '</span>' .
160         '</span>' .
161         '</div>' .
162         '</form>' .
163         '<div_␣id="ssfa_fileup_files_container_' . $uid . '"_class="
            ssfa_fileup_files_container"></div>' .
164         '<span_␣id="ssfa_rf_' . $uid . '"_style="display:none;"></span>' .
165         '</div>' .
166         $clearfix;
167     return $initialize . $form;
168 }
169 }
170 }

```

## B PRÍSTUP NA WEBOVÚ STRÁNKU

### B.1 Povolenia na prihlásenie:

Prihlásenie na vytvorený webový portál pre účely VUT je nastavené na prístup pre členov fakulty, ostatným pre prístup zamietnutý.

### B.2 Ako sa prihlásiť:

Webový portál je dostupný na adrese <http://weswp.utko.feec.vutbr.cz/>. Prihlásenie je realizované prostredníctvom položky „Prihlásiť“ na ľavom bočnom paneli hlavnej stránky webového portálu. Po kliknutí nasleduje presmerovanie na prihlasovaciu stránku VUT, kde sa pomocou VUT konta užívateľ prihlási. Po úspešnom prihlásení sa uskutoční presmerovanie späť na pôvodnú stránku.

### B.3 Administrátorský prístup:

Administrátorský prístup je možný kliknutím na odkaz na ľavom paneli hlavnej stránky webového portálu „Administrátorský prístup“, či linku Administrátorská sekcia s prihlasovacími údajmi:

**Meno:** student

**heslo:** vut123

### B.4 Dočasne vytvorená emailová schránka

Pre účely fungovania zasielania automatických mailov po pridaní nového príspevku bol vytvorený dočasný účet prostredníctvom služby Gmail. Prístup na tento účet je na adrese <http://gmail.com>:

**prihlasovacie meno:** weswp.utko.feec.vutbr.cz

**heslo:** vut123456789

## C OBSAH PŘILOŽENÉHO CD

- Elektronická verzia bakalárskej práce vo formáte PDF
- Zdrojové kódy webového portálu <http://weswp.utko.feec.vutbr.cz/>